PHD

An investigation into the use of digital filtering methods applied to the simulation of continuous communications systems.

Metcalfe, J.

Award date:
1976

Awarding institution:
University of Bath

AN INVESTIGATION INTO THE USE OF DIGITAL FILTERING METHODS APPLIED TO THE SIMULATION OF CONTINUOUS COMMUNICATIONS SYSTEMS

Submitted by J. Metcalfe, B.Sc.

for the degree of Ph. D. of the

University of Bath

1976

COPYRIGHT

J Metcalfe

SUMMARY

This work is concerned with the application of digital filtering methods to the problem of simulation of communications and signal processing systems. The requirements of a generalised simulation scheme are discussed in the light of the extensive literature on digital simulation. Digital filtering methods are compared with numerical integration methods at a conceptual level. The hypothesis is then developed that, for simulation of the type of system under consideration, model design should be based upon accurate simulation of frequency response. The main body of the work then concentrates upon the design of digital filters which accurately model the frequency responses of continuous filters. To this end, digital filter transformation design methods are investigated and methods are developed for simulation-filter design. Digital filtering methods are also shown to be useful for the simulation of function generation elements by providing alias error correction. Non-linear and modulation system elements are discussed, but unfortunately the scope of the work is not sufficient to cover these aspects in great detail.

In addition to the mainstream of the work an interesting side issue arose during the investigation of a filter design method, resulting in the development of a practical computer algorithm for system identification via the Weiner-Lee transform.

This type of work requires extensive use of digital computer aids. Software development is time-consuming and the written work cannot reflect the effort so involved. For this reason an appendix has been included which describes the more important elements of the developed software.

<u>LIST OF CONTENTS</u>

1. INTRODUCTION

Simulation is the building of a model of a device or system. Models are useful in that they provide a means of testing ideas and designs without the complication or expense of building a prototype system, and modelling is widely accepted in engineering. In some fields the models are physical devices, such as small scale versions of the actual system. When a system can be defined mathematically, modelling can proceed from a conceptual viewpoint and the model need not be practically realisable. Such is the type of simulation that can be used for communications or signal-processing systems. If a generalised approach is maintained in the investigation of simulation methods, then one can expect to produce a useful tool for the analysis and design of many systems. The generalised approach will mean that simulation is possible for a wide range of problems, which can be as diverse as:

(i) error measurements in a high-speed digital communications system, with transmission-medium interference; or

(ii) distortion measurements in a low frequency analogue system.

One of the chief functions of a digital computer is to perform arithmetic operations upon numbers, and a sequence of numbers can represent the instantaneous amplitude of a signal. It is conceivable, therefore, that if the correct arithmetic operations are used, sequences of numbers could be converted into other sequences, in such a way that the overall operation models some element in a communications system. Such is the basis of simulation by digital computer.

'Digital filter' is an expression describing a particular type of arithmetic operation, otherwise known as a linear difference equation. The importance of digital filters is that they have similarities with analogue filters in the way that the inputs and outputs are related. These relationships can be such that a frequency-dependent transfer characteristic is produced, hence the term digital 'filter'.

The remainder of this introductory section will deal with certain aspects of the simulation problem by:

(i) investigating the types of model that are required;

(ii) reviewing the literature on digital computer simulation from mathematical and operational viewpoints;

(iii) introducing those concepts of digital filtering which apply to this work.

In this way the scene will be set for an investigation of the suitability of digital filters as simulation aids.

1.1 The System Elements

This work considers simulation methods for communications or signal processing systems. Such systems consist of certain basic elements which thus form essential units in any simulation scheme. In this section these basic 'system elements' are considered. Consideration is given to the nature of the elements and to the methods of elemental specification which a simulation user may wish to employ.

1.1.1 Filtering Elements

A 'filtering element' is a device designed to adapt the spectrum of a signal. In the majority of cases such a device can be defined to have inputs and outputs related by a set of linear, constant coefficient, differential equations. Other filtering devices can often be considered as the combination of an element as defined above and other types of system element.

In practice filtering elements are specified in many ways, often independent of the type of filter to be simulated (whether it is low-pass, high-pass, etc.).

(a) Specification by a transfer function in the Laplace variable 's'. This is probably the most common method used to specify a filter. The definition being:

$$H(s) = \frac{P(s)}{Q(s)}, \qquad 1.1$$

where P(s) and Q(s) are polynomials in 's'. The transfer function H(s) thus specifies the input-output relationship of the device, whose steady state frequency response is given by setting $s = j\omega$.

(b) Specification by a function of the radian frequency variable ω Defined simply by a transfer relationship $F(\omega)$. An example of specification in this way is the Gaussian filter relationship:

$$F(\omega) = e^{-k\omega^2}. \qquad 1.2$$

The difference between this definition and that of definition in the variable 's' is that, filter definition by such functions as Equation 1.2 is often not possible by employing a relationship like Equation 1.1.

(c) 'Piecewise' or 'Band' specification, a common method of designating frequency responses. In this case the frequency response of the filter is split into sections which each represent an essential element in the transfer response. Typically such areas would be known as 'pass', 'stop' or 'transition' bands, each giving the basic action of the device in a specific area of the spectrum.

1.1.2 Function Generators

'Function generation' is taken to define signal producing elements. Signals can be divided into three main categories namely periodic, aperiodic and random functions. All categories of signals will need to be simulated. Function generator specification could be in terms of a function of time or frequency; or a probability density function for random signals.

1.1.3 Modulation Elements

A modulation element is a device in which one signal acts upon another signal to change its phase or amplitude. In practice modulation is achieved by a large variety of devices based upon several prinicples.

For example, amplitude modulation may be implemented by the use of a device with a non-linear transfer characteristic, or by switching methods. Therefore, in addition to considering specifications of idealised modulation elements, the wide variety of practical elements will have to be catered for.

1.1.4 'Non-Linear Elements'

In this case the device definition is in terms of a frequency independent, non-linear signal transfer characteristic. In practical terms a non-linear device will not produce a sinusoidal output when using a sinusoidal input. Specification of non-linear devices is normally undertaken by two basic methods.

(a) Soft characteristics, defined by a power series, for example:

$$y(t) = \sum_{i=0}^{N} a_i \{x(t)\}^i, \qquad 1.3$$

where $x(t)$ is the input signal,

$y(t)$ is the output signal,

a_i are a set of coefficients.

(b) Hard characteristics, usually specifying devices where a switching or clipping operation is involved. For example a device often known as a limiter, is defined:

$$y(t) = \mathrm{sign}(x(t))\ . \qquad 1.4$$

1.1.5 Overall Observations

The foregoing discussion represents the approach to the investigations of simulation methods by indicating the types of devices that will require modelling. The overall approach to the elemental models should be as generalised as possible, such that the final system is not restricted to a small class of problems. It is felt that the major factor in preserving a general approach is to allow a wide range of model specification methods. The discussion so far has indicated how the individual elements may be specified. In addition to these individual differences of specification, consideration must be given to what can be termed 'ideal', 'practical' and 'representative' models. These terms are meant to illustrate the range of types of specification that could be encountered. The terms can be illustrated by the example of the specification of an amplitude modulator. An 'ideal' model could be a straightforward multiplier. A model 'representative' of nonlinear modulators could be a square-law. Whereas the 'practical' nonlinear amplitude modulator would have a full power series transfer characteristic.

Thus it is felt that strict methods of elemental specification cannot be made at this stage. As many specification methods as possible should be considered if a generalised approach is to be maintained and if a simulation system users requirements are not to be pre-empted.

1.2 A Review of Digital Simulation Methods

The use of digital computers for the simulation of analogue systems is a wide field, and over recent years a large amount of literature has been published. The literature concerning simulation systems tends to fall into several main areas:

(i) 'Discrete simulation', this area is of no direct interest to the analogue simulation problem, in that it concentrates upon 'discrete change' simulations. An example of a discrete change simulation would be say, a queueing problem.

(ii) Simulations designed around a particular problem. It is true that the systems considered herein are specialised to the extent that only communications systems are being considered. The approach to these simulations, however, is to be as generalised as possible. Therefore the specialised simulations in the literature will only be of use where they help with the simulation of particular system elements.

(iii) 'Continuous' (Analogue) simulation methods. The area of generalised analogue simulation methods forms the main area of interest in the literature. These methods will not necessarily form the basis of usable simulation schemes for communications systems. The importance of these methods is in their historical development, their organisation and the structure of their basic elements.

(iv) Existing methods for the simulation of communications systems. Generalised methods for simulation in this specific area are not found extensively in the literature. Those methods that are reported are felt to be rather limited in their application.

(v) Fast Fourier Transform (F.F.T.) methods are useful for the analysis of communications systems. The FFT is also used in the simulation area. The essential difference between using the

FFT and other methods is that the FFT processes the signal data in blocks of samples. Now, it is the intention of this work to consider simulation methods which operate from a 'sample-by-sample' approach. Therefore FFT simulation methods will not be considered deeply, as they are outside the scope of the work reported herein. The FFT will be used, however, as a valuable tool for the analysis and design of elemental models.

1.2.1 A Brief Chronology

Before the wide availability of digital computers, simulations were usually constructed using analogue computers. The interest in digital computer simulation developed with the spread of digital machines. Due to the previous use of analogue computers, early digital simulators tended to model the analogue computer components. The initial work in this area is reported by Selfridge (Ref. 51), who developed algorithms for the simulation of analogue computer components. These 'analogue computer' methods require a user to specify his problem in terms of its basic differential equations. The simulation was then operated as though an analogue computer were being used. Much literature was produced in the 1960's about simulation, a large proportion of it concerning these 'analogue computer' methods.

The earlier elements of this work (e.g. References 18, 27 and 31) required the user to supply analogue computer patching information to the digital machine. The 'patching' type of simulation system is often referred to as a 'block orientated system' as the analogue computer components are treated as items in a block diagram.

Later work showed a move towards defining the simulation by its equations, in the form of a programming language (e.g. References 8, 36, 55 and 57). Such languages were often very similar to the FORTRAN programming language. The language orientated systems show a greater flexibility, particularly where a mixture of the simulation and programming languages is allowed. Contemporary developments also provided schemes for simulation which were used as subroutines in association with a standard programming language. Such organisational features will be discussed in greater detail later.

Few simulation languages apply specifically to the simulation of communications systems in a general way. Those that are found in the literature again fall into two areas. In much the same way as for the 'analogue computer' methods, the two areas are 'block orientated' and 'language orientated'. BLODIB (Ref. 23) is of the block type. KAIRO (Ref. 48) and SYSTID (Ref. 19) are of the language type. The communications related simulation methods have been a later development than the 'analogue computer' types. Details of the specific cases will be given later.

1.2.2 The 'Analogue Computer' Methods

This section deals with those methods which are concerned purely with the simulation of analogue computer components., (References 8, 18, 27, 31, 36, 51, 55 and 57). The basis for these methods is the simulation of the analogue integration element by numerical integration methods. Practical numerical integration methods are reviewed with respect to simulation in References 1, 20 and 61. Without going into great detail, numerical integration methods can be briefly expounded upon. A more detailed treatment of numerical integration can be found in Reference 26.

Numerical integration methods fall into three basic areas:

(i) Difference equation methods, where the integration operator is simulated by a linear difference equation. An example of a linear difference equation for an input sample stream x(kT) and an output y(kT) is:

$$y(kT) = h\,x(kT) + y(kT-T)\,, \qquad 1.5$$

h being a constant and T a time interval.

Linear difference equations are often expressed in the form of their Z transform transfer relationships. The use of Z transformations for linear difference equations is well known and is based upon the relationship:

$$Z = e^{sT}\,, \qquad 1.6$$

where s is the Laplace variable.

Expressing the linear difference Equation 1.5 in terms of its Z transform transfer relation gives:

$$H(Z) = h\,\frac{Z}{Z-1}\,. \qquad 1.7$$

For numerical integration purposes if

$$h = T \qquad 1.8$$

then Equations 1.5 and 1.7 represent the relationship of a numerical integration method known as the Euler method.

Another typical linear difference equation integration method is represented,

$$H(Z) = \frac{h}{2}\,\frac{Z+1}{Z-1} \quad 1.9$$

and is known as the trapezoidal corrector. These types of numerical integration methods are used by operating the transformation:

$$\frac{1}{s} \Rightarrow H(Z) \quad 1.10$$

(ii) Predictor-Corrector methods, which use a pair of linear difference equations. One of the equations estimates the value of the integrated output at the next sample point (i.e. it predicts). The other equation (the corrector) is then applied to the input and predicted data to arrive at a second result. The difference between the two results is then used to arrive at a further (corrected) result. In practical simulation systems the difference between the predictor and corrector outputs is used as a measure of error (as even the corrected result is not absolutely accurate). The error measurement is normally used to reduce the sampling interval when defined limits are exceeded. An example of a simple predictor-corrector pair is, predictor:

$$y_{n+1} = y_{n-1} + 2h\,x_n \quad 1.11$$

and the corrector:

$$y_{n+1} = y_n + h\,\frac{x_{n+1} + x_n}{2} \quad 1.12$$

Gagné and Baxter (Ref. 20) review numerical integration methods and give several examples. A more complex predictor-corrector method could use the equations:

$$H(Z) = \frac{1}{24} h \frac{55Z^3 - 55Z^2 + 37Z - 9}{Z^3(Z-1)} \qquad 1.13$$

known as the 'Adams-Bashforth' predictor, and:

$$G(Z) = \frac{4}{90} h \frac{7Z^4 + 32Z^3 + 12Z^2 + 32Z + 7}{Z^4 - 1} \qquad 1.14$$

known as the 'Milne' corrector. Much of the discussion in the literature centres around the problems of accuracy, stability and starting the integrators. The accuracy of the simulations is checked by the error values found from the difference between the predicted and corrected results. If this value is too high then the integrators are re-started with reduced step size. The stability problem is concerned with the fact that certain integration methods are not suitable for certain problems. Thus the integration method used in practice must be selected to suit the type of problem. The starting problem is concerned with the initial values used in the linear difference equations. In practice two methods are used for starting:

(a) by using the differential equation and its derivatives to form a Taylor expression:

$$y_1 = y_0 + h\, y_0' + \frac{h^2}{2} y_0'' + \frac{h^3}{3!} y_0''' \ldots \text{ etc.} \qquad 1.15$$

which can then be used to find the initial values;

(b) by the use of 'Runge-Kutta' methods.

(iii) 'Runge-Kutta' methods, (Ref. 26), these methods are basically different from the difference equation methods, in that no historical data samples are used. The most commonly used Runge-Kutta method is defined as follows:

given the differential equation:

$$\dot{y} = f(x,y) \tag{1.16}$$

then calculating:

$$\begin{aligned} k_1 &= hf(x_n, y_n) \\ k_2 &= hf(x_n + \tfrac{h}{2}, y_n + \tfrac{k_1}{2}) \\ k_3 &= hf(x_n + \tfrac{h}{2}, y_n + \tfrac{k_2}{2}) \end{aligned} \tag{1.17}$$

and $k_4 = hf(x_n + h, y_n + k_3)$

the next integrated value can be found,

$$y_{n+1} = y_n + \tfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \ . \tag{1.18}$$

This process effectively involves the estimation of four slope values, the weighted average of these slopes then provides the advancement to the next point. The variants of this method concern the use of different sample positions for the estimates of the slope and hence different weights. This method can be looked upon as a curve fitting exercise, fitting the samples to the differential equations.

1.2.3 Methods of Simulation for Communications Systems

The 'analogue computer' simulation methods can obviously be applied to communications systems. The application is likely to be complicated by the need to break-down any problem into its differential equations. Most of the literature specifically concerning communications system simulation is of a specialised nature, usually concentrating on a particular problem. Some of the reported work is of more general application.

Sakai and Niimi (Ref. 48) have devised a simulation system based upon a set of 'unit circuits'. The 'unit circuits' form a set of building blocks for the construction of a simulation. The allowable units are rather restricted, for example filters of a maximum order of two are allowed (therefore higher order cases would require splitting). The simulation language (known as KAIRO) is used by supplying the interconnections of a block diagram made up from the available unit circuits. Unfortunately Sakai and Niimi give no details of the method of modelling employed for the unit circuits. However, though one cannot be certain, it is possible to infer from the paper that the basis of the modelling is that of the 'analogue computer' methods. The difference being that the language is structured to be relevant to a communications system orientated user.

Golden (Ref. 23) and Kuo (Ref. 37) are concerned with a simulation system known as BLODIB, developed by Bell Telephone Laboratories. BLODIB was developed specifically for the simulation of sampled data systems. Simulation of analogue components is, therefore, not easy. For example, filters may be simulated by the use of digital filter blocks, however the user would have to arrange his own digital filter design.

Probably the most applicable simulation system for the purposes of interest herein, is the system known as SYSTID (Ref. 19).

The SYSTID system is very comprehensive, allowing all of the elements mentioned in Section 1.1, and enabling 'low-pass equivalent' simulation. The filtering components are dealt with by the use of digital filters. However the SYSTID simulator is somewhat restricted, in that only Bilinear Z Transform designs (see later) of classical filter prototypes are allowed.

1.2.4 Typical Simulation Problems

The 'analogue computer' methods are open to a wide range of simulations, and many examples are to be found in the literature. However, one particular problem is often reported and seems to have become a standard test for this type of simulation. This problem is that of simulating a pilot ejecting from an aircraft. The object of the simulation being to determine the ejection parameters, such that the pilot's trajectory ensures that he does not strike the tail of the aircraft. Another typical problem is the simulation of the mass-spring-damper systems used as aircraft arresting equipment on aircraft-carriers. Such typical simulation problems illustrate the most profitable usage of the 'analogue computer' methods. That is, for problems where the explicit simulation of differential equations is important.

The communications system simulators naturally give examples of simulations of a fundamentally different nature to those mentioned above. For example, in reporting the KAIRO system (Ref. 48) the authors consider the simulation of a system to produce the zero crossing wave of a voiced sound. The literature on the SYSTID (Ref.19) system gives two examples of the use of the simulator. The first example is a very simple case giving the step response of a filter. The second example is more complicated, being the simulation of a pulse-coded modulation/phase modulation system used on the APOLLO spacecraft programme. These latter examples are in line with the type of system one would expect to simulate with any scheme orientated towards communications systems.

1.2.5. How the System Elements are Simulated

In this section the discussion will concern the way in which the simulation schemes reported in the literature could, or do, handle the 'system elements' as detailed in Section 1.1.

1.2.5.1 Filtering Elements

The 'analogue computer' methods require the filter to be decomposed into its differential equations. The differential equations are then simulated as though an analogue computer were being used. The KAIRO (Ref. 48) system allows direct specification of filters of first and second order. Therefore, with KAIRO, high order filters would have to be reduced to combinations of second and first order elements. Both the BLODIB (Ref. 23) and SYSTID (Ref. 19) systems simulate filtering elements by using digital filtering algorithms. With BLODIB one would have to supply the pre-designed digital filter coefficients. With SYSTID the filters are specified by several classical filter shapes only, no direct specification by transfer function being allowed. For example, one could specify an eighth order, band-pass, Butterworth filter of defined bandwidth and centre frequency.

Clearly all of these methods could be extended to allow a more flexible definition of filtering elements. For example with SYSTID, in addition to what one may term definition by 'type', transfer function definitions could be allowed.

1.2.5.2 Function Generation

The 'block orientated' type of 'analogue computer' simulation systems allow only a defined set of elements to be used. In most cases the elemental set includes simple function generation blocks, often only sinusoids. Therefore, if it were necessary to generate a square wave, say, then one would have to first generate a sine wave and pass it through a comparator. Functions could, of course, be generated by the use of analogue computer techniques, that is by setting up the differential equations of the required signal.
The later 'analogue computer' simulation systems allow a wider range of functional blocks. Where a mixture of simulation and standard programming languages is allowed the function generation problem becomes much easier as the full range of computer facilities is available. Of the simulation systems designed specifically for communications systems SYSTID (Ref. 19) offers the most comprehensive range of function generation elements. The elements allowed by the SYSTID system cover the whole range of functions discussed in Section 1.1.2.

It is important to note that, of all of the items of literature studied, no simulation system has been found which considers the ramifications of sampling theory when generating non-bandlimited functions.

The procedure normally used to overcome alias problems due to sampling is simply to increase the sampling rate. This action must increase the amount of computation necessary to effect the simulation.

1.2.5.3 Modulation Elements

In the 'analogue computer' simulation systems no special elements are available for the simulation of modulation devices. Of the other simulation systems studied only SYSTID (Ref. 19) deals clearly with modulation elements. SYSTID allows definition of modulation and demodulation components as standard blocks of the language. The language also allows complex low-pass equivalence simulation in the case of narrow-band systems. The modulation elements allowed by the SYSTID system are, however, idealised and no scope for the simulation of practical modulation components exists.

1.2.5.4 Non-Linear Devices

In all of the simulation systems investigated in the literature non-linear devices are allowed. SYSTID (Ref. 19) is the only simulation system which shows a clear distinction between 'hard' and 'soft' characteristics. The remainder of the systems only specify the 'hard' type of characteristics, although it is clear that soft characteristics could be simulated by networks of summation and multiplication elements. In all of the non-linear simulations found the device characteristic is simulated directly, no account is taken of the alias effects that can result from non-linear elements in digital systems.

In addition to the type of non-linear devices discussed in Section 1.1.4 other system non-linearities can be considered. For example the 'analogue computer' literature considers extensively the simulation of non-linear differential equations. Such equations are more amenable to simulation by the 'analogue computer' methods. These devices are not a regular feature of communications systems. However, it is advisable to consider the retention of elements of the 'analogue computer' methods in any simulation scheme devised, in order to preserve generality.

1.3 The Ergonomics of System Simulation

The methods of organisation of system simulation have been studied but no direct work has been done in this area. This study gives clear lines of approach and it is thus worth reporting.

The development of the organisation of simulation systems can be observed from the literature as reported in Section 1.2.1. This development is particularly clear from the 'analogue computer' methods. The earlier methods (e.g. References 18, 27 and 31) involved the user in supplying patching information, just as if he were using an analogue computer. Such a system is inflexible as the only simulable elements are those originally allowed for in the simulation system design. Later developments (References 8, 36, 55 and 57) allowed more flexibility by the use of simulation 'languages'. These languages often being similar to and compatible with standard programming languages such as FORTRAN. Flexibility is inherent in such a system as extension of the system elements is allowed by normal programming methods.

Of the communications system simulators the more important ones (References 19 and 48) use a language structure compatible with a standard programming language.

The study of the literature enables a clear statement to be made of the desirable features of the organisation of a simulation system. It is felt that these features will give a viable system of wide usage and good flexibility.

(a) 'A simulation language is essential'

Simulation languages, similar to mathematical programming languages, are used by the majority of the more advanced simulation systems.

(b) 'System extension must be possible'

It is unlikely that a simulation system designer will cater for all possible requirements of the users. Therefore it must be possible to extend the simulation system.

(c) 'Flexible model definition is required'

In Section 1.1 various ways of defining the system elements were discussed. If the definition of a system element is allowed in as many ways as possible, a good flexibility will result.

(d) 'Totally automatic design'

Once the system to be simulated has been defined, the resultant construction of the simulation should be automatic, subject to error criteria.

(e) 'Two-stage simulation is desirable'

With a 'two-stage' simulation the performance of the simulation is clearly separated into 'compile' and 'run' functions. The 'compile' function involves the assembling of the simulation language into a programming language (say FORTRAN). The 'run' function is concerned with the execution of the program which results from the 'compile' stage. Reference 48 describes a system which works on this basis.

(f) 'No familiarity of the user with computer methods should be necessary'

Earlier elements of the 'analogue computer' simulation methods (for example Reference 27) state that the user should be familiar with computer systems. The later development of simulation Languages show a move away from these ideas. Reference 55 specifically states that users should not be required to be familiar with computer systems.

1.4 The Philosophy of the Use of Digital Filters

Most of the numerical integration algorithms described in Section 1.2.2 are transformations from the 's' to the Z^{-1} plane. The transformations are expressed in the form of transformations of the integration operator, that is,

$$\frac{1}{s} \Rightarrow H(Z^{-1}) \; . \qquad 1.19$$

Such transformation methods relate to the field of digital filtering, in that certain methods of digital filter design are transformations of the form of Equation 1.19. For instance, the trapezoidal corrector formula (Equation 1.9) is identically the Bilinear Z transform (see later). Other commonly used transformations used for digital filter design are not directly applied to the s variable, and therefore cannot be considered in terms of the integration operator. However, such transformation methods may well be valuable in terms of system simulation. In addition to transformation methods of digital filter design other design methods exist, which may possibly be profitable when used for simulation purposes.

To summarise, the discussion up to this point indicates that numerical integration methods result in digital filter type structures by substitution transformation for the 's' variable. Therefore, it is intended to investigate the use of digital filter design methods for simulation purposes. Many digital filter design methods are not applied by direct substitution for the 's' variable and are therefore not considered by the majority of the literature, which is concerned with the simulation of the integration operator.

Consideration must now be given to the performance requirements of any simulation scheme. For good performance the results of a simulation should be as accurate as possible and the speed of operation should be as fast as possible.

(i) Accuracy

The assessment of accuracy in a simulation is rather dependent upon the desires of the user. In a perfect model all responses to excitation would precisely match the responses of the real system. That is, for the simulation of a communications system both time and frequency responses would be accurately modelled. However in the work considered herein simulation is to be enacted in a digital environment. Digital simulation means that all signals are represented by a series of samples. Sampling theory shows that any sequence of samples represents an infinite class of signals. The sampling theorems also indicate that, for unambiguous reproduction of the original signal, the signal must be band-limited before sampling. The width of the required band-limiting is ω_s (the sampling frequency)[1]. Therefore, for digital simulation any real signal represented by a set of samples must be band-limited for accuracy in the frequency domain. If, however, the signal is accurately represented in the time domain, without band-limiting then no reliance can be placed upon the resultant frequency response. It is evident therefore that, in the absolute sense, time and frequency responses are incompatible in their accuracy requirements, except in the limiting condition when the sampling frequency approaches infinity. The accuracy problem is thus dependent upon which domain one chooses to concentrate ones effort. The 'analogue computer' simulation systems tend to concentrate their effort upon accuracy in the time domain. This tendency is often justified, in that many of the models considered are only dependent upon one type of excitation (for example, in the pilot ejection simulation the only excitation is the ejection force). For communications systems however one is often interested in the response of a system to a wide range of excitations. Thus, and considering sampling theory, when generating driving functions for a model time domain accuracy is dependent upon one question. The question is:

'What is the highest frequency of interest?'

1 For low-pass systems this means a cut-off frequency of $w_s/2$

If a signal is sampled such that this highest frequency is unambiguously represented then subsequent accuracy (in both domains) is only dependent upon the accuracy of the frequency response of the elemental models. It is true that many typical signals are not bandlimited in practice (for example, square pulses), however in these cases absolute accuracy in both domains is not possible when digital simulation is used. The accuracy of simulation for such signals can only be considered in the light of the above question. The above observations become more clear when one considers digital filter design using the 'standard Z transform' (see later). This transform gives an exact simulation of the sampled impulse response of a filtering element when the digital version is excited by a unit pulse sequence. However the frequency response of the original filter is not accurately simulated over the Nyquist range.

(ii) Operating Speed

The time required to run a digital simulation is dependent upon both the sampling rate and the complexity of the model. Most recursive digital filter design methods translate from the 's' to the Z^{-1} planes such that the order of the digital model is similar to that of the original filter. Equations 1.13, 1.14, 1.17 and 1.18 show that when numerical integration methods are used the effective order of the simulation is often much greater than that of the original system. This is especially true when predictor-corrector methods are used as such methods employ two linear difference equations. Therefore, when using 'analogue computer' methods the order of the model can easily be several times the order of the original system.

Sampling rate considerations are dependent upon the highest frequency of interest in the particular simulation. Giese (Ref. 21) states that typically sampling rates are 10 to 30 times the highest frequency of interest for numerical integration methods. However, where predictor-corrector methods are used, sampling rates are difficult to quantify due to the ability of these methods to adjust the sampling rate with respect to error criteria. From sampling theory it is known that sampling rates need only be twice the highest frequency of interest, providing that the signals are band-limited.

The above considerations lead to criteria for the economic design of simulation elements. Firstly the sampling rate should be kept as low as possible (ideally twice the highest frequency of interest). Secondly that the order of the model should be as low as possible. It should be noted that in practice a 'trade-off' may exist between the two criteria, particularly when the effort of designing the model is considered.

The foregoing discussion leads to the objectives of this work. Those objectives being to investigate the use of digital filter design methods for producing simulations. It is felt that digital filter methods will aid economic simulation system design by enabling low-order models to be used. The design of the digital filters involved is to be studied with respect to the accuracy of the resultant model in the frequency domain. This frequency response accuracy is important in that it is the only way of unambiguously considering the signals present, and that it holds the possibility of using low sampling rates.

1.5 The Approach to Digital Filter Design

Section 1.3 discussed the arguments in favour of considering the use of digital filter methods for designing simulation models. The basis of the filter design is to be an attempt to create filters which model the frequency response of the original device.

Digital filter design will be considered from several bases, namely:

(i) transformation from the 's' to the Z^{-1} plane;

(ii) the use of a unit pulse response;

(iii) the use of frequency responses, for both direct design and for deriving unit pulse responses for use as design bases.

It is important that generality be preserved when approaching the problem of simulating filtering elements. Therefore methods of digital filter design will be developed and analysed in a way which is independent of the type of filter. That is, independent of the filter being low-pass, high-pass etc.

2. DIGITAL FILTER DESIGN METHODS

This chapter is aimed at investigating the use of existing digital filter design methods. The object of this investigation is to analyse the responses of the resultant digital filters. This analysis will be based on a comparison of the frequency responses of the original 'continuous' device and the digital model. The design methods fall into two broad areas based upon the specification of the original filter.

(i) Transformation methods, which transform a 'continuous' transfer function to produce a digital domain transfer function.

(ii) Methods based upon the design of filters to have specific types of frequency responses which do not have a strict transfer functional definition.

The literature concerning digital filter design is extensive but is mainly concerned with the design of filters for practical filtering operations. Such interests mean that much effort is devoted to problems such as word length and quantisation. The requirements for digital simulation are not the same as those for practical filters, and as a result very little of the literature is of direct relevance to simulation filter design.

2.1 Transformation Methods for the Design of Recursive Digital Filters

(References 6, 22, 24 and 37)

Transformations are defined by an operation from a function of 's' to a function of Z^{-1}:

$$P(s) \Rightarrow Q(Z^{-1}). \qquad 2.1$$

The numerical integration methods discussed in Section 1.2 were shown to be, in many cases, transformations of the form:

$$\frac{1}{s} \Rightarrow H(Z^{-1}) . \qquad 2.2$$

Practically this transformation is implemented by a simple substitution for the variable in the original transfer function, (i.e. a conformal mapping). Consider the function:

$$G(s) = \frac{A(s)}{B(s)} \qquad 2.3$$

transforming gives:

$$V(Z^{-1}) = \frac{A(1/H(Z^{-1}))}{B(1/H(Z^{-1}))} . \qquad 2.4$$

Other types of transformations to the Z^{-1} plane are not implemented by a simple mapping. They require G(s) to be broken down into sub-functions each being transformed. These alternative methods are not analogous to the simulation of the integration operator and may, therefore, throw new light into the area of digital simulation. Three transformation methods are commonly used to design digital filters. Following is a brief explanation and discussion of these methods.

2.1.1 The Standard Z Transform (References 22, 24, 30, 32, 37 and 62)

This transformation is based upon a defined equivalence of the impulse and the digital unit pulse sequence.

Consider the linear difference equation:

$$y_n = L\, x_n + K\, y_{n-1} \, . \qquad 2.1$$

Also consider the case where the input is a unit pulse sequence:

$$x_n = 1, 0, 0, 0, \text{etc.} \qquad 2.2$$

$$\text{for } n = 1, 2, 3, 4 \text{ etc.}$$

Then the output sequence becomes:

$$y_n = L, LK, LK^2, LK^3 \text{ etc.} \qquad 2.3$$

Now let $K = e^{-aT}$ and $p(nT) = y_n$, T being the sampling interval.

Then:

$$p(nT) = L, Le^{-aT}, Le^{-2aT}, \text{etc.} \qquad 2.4$$

Which is clearly equivalent to the sampled impulse response of the continuous transfer function:

$$F(s) = \frac{L}{s + a} \qquad 2.5$$

the impulse response of which is:

$$f(t) = L\, e^{-at} \, . \qquad 2.6$$

Defining the notation '*' to indicate the sampled version of a signal (i.e. $f(nT) = f^*(t)$) and considering the Laplace transform of Equation 2.6:

$$F^*(s) = \int_0^\infty L\, e^{-anT}\, e^{-st}\, \delta(t - nT)dt \qquad 2.7$$

Remembering that $f^*(t)$ only exists at the sample points:

$$F^*(s) = L\left\{1 + e^{-aT}e^{-sT} + e^{-2aT}e^{-2sT} + \text{etc}\right\} \qquad 2.8$$

i.e. $$F^*(s) = L \sum_{k=o}^{\infty} e^{-kaT} e^{-sT} \qquad 2.9$$

then: $$F^*(s) = \frac{L}{1 - e^{-aT}e^{-sT}} \qquad 2.10$$

assuming that Equation 2.9 is convergent (a reasonable assumption as only stable systems are of interest).

Defining,

$$Z^{-1} = e^{-sT} \qquad 2.11$$

(in concurrence with Z transform theory) the transformation from the s to the Z^{-1} plane becomes:

$$\frac{L}{s+a} \Rightarrow \frac{L}{1 - e^{-aT}Z^{-1}} \qquad 2.12$$

The assumption associated with Equation 2.10 means that stable systems have poles outside the unit circle in the Z^{-1} plane.

Expanding the denominator of Equation 2.10 gives:

$$F^*(s) = \frac{L}{1 - \left\{1-(a+s)T + \frac{(a+s)^2T^2}{2!} - \frac{(a+s)^3T^3}{3!} \text{ --- etc.}\right\}} \qquad 2.13$$

Now the limit as $T \to 0$ (that is, as the digital system approaches the continuous system) gives:

$$\underset{\lim T\to 0}{F^*(s)} = \frac{L}{(s+a)T} \qquad 2.14$$

Therefore for dimensional correctness Equation 2.12 becomes:

$$\frac{L}{s+a} \Rightarrow \frac{LT}{1 - e^{-aT}Z^{-1}} \qquad 2.15$$

and the unit pulse sequence should be considered as $\frac{1}{T}$, 0, 0, 0, etc. This transformation is known as the 'Standard Z Transform'. The analysis can be extended to cases of multiple poles where:

$$\frac{A}{(s+a)^m} \Rightarrow T^m \left\{ \frac{(-1)^{m-1}}{(m-1)!} \frac{\partial^{m-1}}{\partial a^{m-1}} \frac{A}{1 - e^{aT}Z^{-1}} \right\} \qquad 2.16$$

(see Reference 22).

The effects of this technique for the design of digital filters relate closely to the implications of the sampling theorems. Thus alias distortion of the frequency response results where, as in most cases, the impulse response of the original device is not band-limited to the Nyquist frequency[1]. This is evident from the fact that the impulse response is modelled exactly in its sampled form. The alias effect can be visualised as the overlapping of the non-band-limited image spectra, centred on the sampling frequency and its harmonics. The resultant distortion of the frequency response can be seen in the examples to be presented later.

2.1.2 The Bilinear Z Transform (References 22, 24 and 37)

The problem of alias distortion discussed in the previous section is particularly severe where the original frequency response does not have a limited pass-band (e.g. high-pass and band-elimination devices). This problem can be, to some extent, avoided by the use of an initial transformation to compress the whole of the 's' plane into a strip which does not extend beyond the Nyquist range. When the transformation to the Z^{-1} plane is then applied to the new compressed 's' plane a new overall transformation exists. This overall

1. Herein Nyquist frequency means half of the sampling frequency.

transformation maps the whole of the 's' plane imaginary axis onto the unit circle in the Z^{-1} plane. A transformation which accomplishes this procedure is a bilinear transform, i.e.:

$$s \Rightarrow \frac{a + bZ^{-1}}{c + dZ^{-1}} \quad . \qquad 2.17$$

The conditions for the mapping of the s plane imaginary axis onto the unit circle into the Z^{-1} plane are:

for $s = o \quad Z^{-1} = 1$,

and $s = \pm j\,\infty \quad Z^{-1} = -1$.

Using these conditions Equation 2.17 becomes:

$$s \Rightarrow k\left(\frac{1 - Z^{-1}}{1 + Z^{-1}}\right) \qquad 2.18$$

This transformation applies a non-linear frequency scale distortion between the domains such that:

$$\omega' = k \tan \frac{\omega T}{2} \qquad 2.19$$

where ω and ω' are the frequencies of corresponding response points in the digital and continuous domains respectively. That is, for the digital response 'd' and the continuous response 'c',

$$d(\omega) = c(\omega'). \qquad 2.20$$

The extent of the distortion is dependent upon the value of k. Usually k is selected such that $\frac{d\omega}{d\omega'}$ is unity at $w = o$, giving low distortion at low frequencies.

The resultant transformation is:

$$s \Rightarrow \frac{2}{T}\left(\frac{1 - Z^{-1}}{1 + Z^{-1}}\right) \qquad 2.21$$

and is known as the Bilinear Z Transform. The effects of this transformation and the extent of the frequency response distortion can be seen in the examples to be presented later.

2.1.3 The Matched Z Transform (References 24 and 29)

From the previous two sections it is seen that:

(i) the application of the Standard Z Transform moves only the poles of a system in a one-to-one correspondence between two planes;

(ii) the Bilinear Z Transform moves both the poles and zeros in a one-to-one correspondence, but introduces distortion of the root positions with respect to the frequency variable.

The Matched Z Transform is designed to perform a one-to-one mapping of roots, whilst maintaining a linear relationship from plane-to-plane. The basis of the transformation is similar to that of the Standard Z Transform. The Matched Z Transform is therefore defined as:

$$s + a \Rightarrow 1 - e^{-aT}Z^{-1}, \qquad 2.22$$

where the root $s + a$ applies to both poles and zeros.

2.1.4 Examples

Four sets of examples are given of the results of applying the Z transformation methods. The frequency responses of the resultant digital filters are each compared to the original continuous frequency response. The figures also show the pole-zero diagrams of the filters. The responses used in the examples were selected to demonstrate cases of both success and failure of the design methods. In many of these (and later) examples it is seen that the frequency response comparisons extend well into the stop-band. Such comparisons are relevant because in certain types of systems stop-band performance is important (for example, a simulation user may wish to analyse adjacent channel interference problems). The contents of the figures are listed below.

(i) Figures 2.1 to 2.4 show the results for a fourth order Chebyshev filter. The filter cut-off frequency is 1 rad./sec. sampling frequency 10 rads./sec. This example shows a low-pass filter where the cut-off frequency is low with respect to the sampling frequency.

(ii) Figures 2.5 to 2.8 demonstrate a case where the cut-off frequency of a low-pass filter is higher in the Nyquist range. The characteristic is a third order Butterworth filter of 4 rads./sec. cut-off. This relatively high cut-off frequency is meant to illustrate the problems that arise when low sampling rates are used.

(iii) Figures 2.9 to 2.12 show the results for a sixth order, Butterworth, band-pass filter of bandwidth 1 rad./sec. and centre frequency 1 rad./sec. This example shows the effect on responses which have low gain at both ends of the spectrum.

(iv) Figures 2.13 to 2.16 demonstrate the response effects of the transformations when the prototype filter has appreciable gain above the Nyquist frequency. This example is of a third order, high pass, Butterworth filter of cut-off frequency 1 rad./sec.

All examples use a sampling frequency of 10 rads./sec. The response curves in each graph are marked 1 and 2. 1 indicates the continuous response and 2 the digital response. Some computational errors are evident on one or two of the graphs. These errors only occur where very low gains exist at very low frequencies. It is believed that these problems are due to rounding errors in calculating very low values of the complex exponential.

2.1.5 Discussion

The examples show clearly the effect of the three transformations in the frequency domain. The pole-zero diagrams show the basis for the differences in the effects of applying each of the transformations. These frequency response differences are seen to be due to the way in which the transformations place zeros in the Z^{-1} plane, (allowing for the warping effects of the Bilinear Z Transform). The examples also indicate that, for different filters, different transformation methods can give superior results. In the following discussion each of the transformation methods will be considered separately.

(a) For the Standard Z Transform

When the Standard Z Transform is used, finite 's' plane zeros are mapped into the Z^{-1} plane such that the relationship with the frequency variable is changed. The effect upon finite 's' plane zeros is particularly well illustrated where zeros exist at s = o, that is zeros do not result at $Z^{-1} = 1$. (Figures 2.9, 2.12, 2.13 and 2.16). Infinite 's' plane zeros produce zeros in the finite area of the Z^{-1} plane. The use of infinite s plane zeros can have beneficial effects. For instance, consider example (i). In this case the prototype filter is 'all pole'. For all pole filters the Standard Z Transform (Figures 2.1 and 2.4) produces finite Z^{-1} plane zeros, while the Matched Z Transform (Figures 2.3 and 2.4) does not, whereas all other roots are in identical positions.

(b) For the Bilinear Z Transform

The frequency response warping effects are clear in the examples. (Note: in practice the effects of warping can be minimised by pre-warping critical points of the original response). The other effect of the Bilinear transform is that infinite s plane zeros are mapped such that zeros lie at $Z^{-1} = -1$ (the Nyquist frequency).

(c) For the Matched Z Transform

The Matched Z Transform maps the roots of the 's' plane transfer function in a one-to-one correspondence. Infinite s plane zeros are ignored by the Matched Transform. The Z^{-1} plane frequency variable ($e^{-j\omega T}$) exists only on the unit circle, and therefore does not approach infinity. Thus, where infinite 's' plane zeros exist the gain of the Matched Z digital filter is greater than that of the original filter.

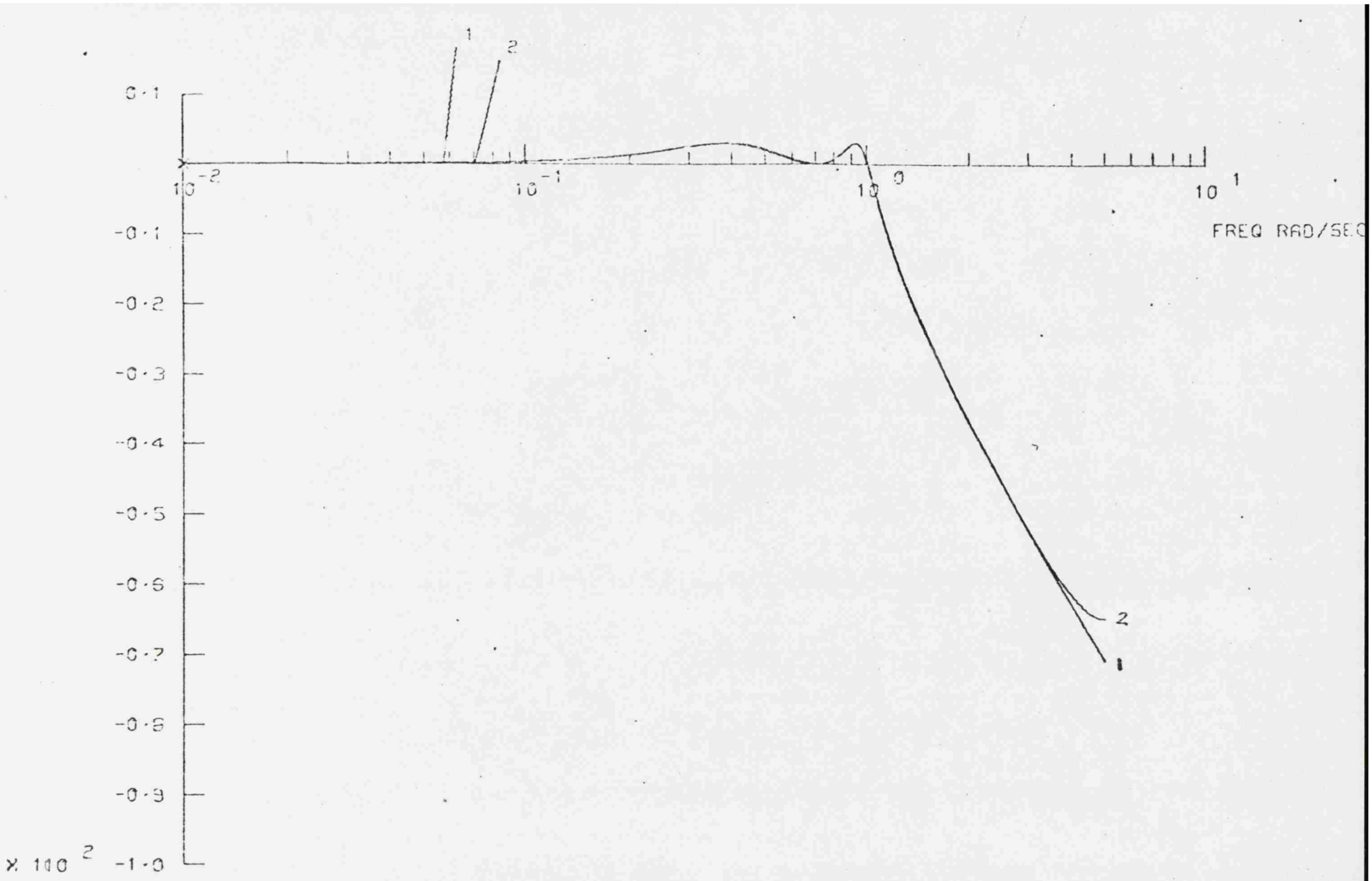
To conclude, it is seen from the examples that the three transformation methods studied produce differing frequency response effects. The differing effects observed seem to be due to the way in which Z^{-1} plane zeros are produced. Further aspects of zeros in the Z^{-1} plane will be discussed later.

2.2 Other Methods of Response Specification

Digital filter design by 's' to Z^{-1} plane transformation was discussed in the previous section. Other methods of specifying filter responses were discussed in Section 1.1. In Section 1.1 it was stated that a simulation system user may wish to specify a response by its essential characteristics, rather than by a specific continuous transfer function, (functions of the frequency variable and 'piecewise' specifications were considered). Such areas of digital filter design are widely reported (References 13, 17, 35 and 39 are typical of this work). Much of the reported work concentrates upon producing filters for a practical digital filtering environment. These 'practical' cases consider such elements of design as word length, quantisation problems and arithmetic significance. These problems are not necessarily of importance in the design of digital filters for simulation purposes.

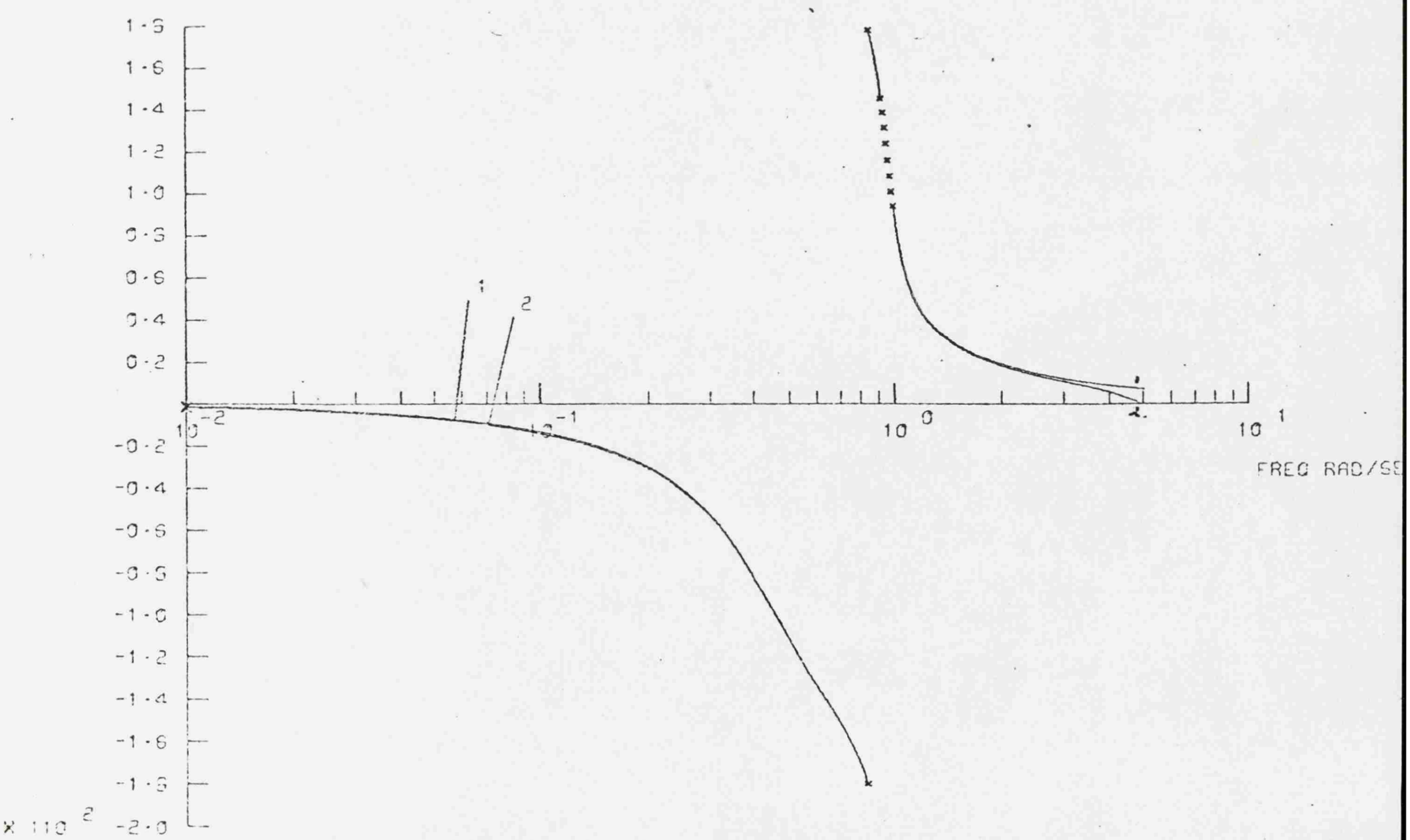
The design of digital filters for simulation purposes requires a different approach to that discussed above. A sensible approach is that used by Freeman et. al. (Ref. 19) who recommend responses specified by their type (for example maximally flat, equiripple, etc.). In this case the filters are first designed in the 's' plane and digitised by the use of the Bilinear Z transform. Some other useful approaches are found in the literature, namely the digital plane filter transformation methods of Constantinides (Ref. 11) and Broome (Ref. 9). These transformations use low-pass digital filter prototypes to produce other filter shapes. For the production of band-pass digital filters this work is reviewed by Martin and the author (Ref. 41) (see Appendix B).

Other than in the work discussed so far no direct effort has been applied to designing digital simulation filters whose responses are specified as discussed here. Later sections develop digital filter design methods which may well have application in this area.



DDB GAIN 1-CONTINUOUS 2-DIGITAL

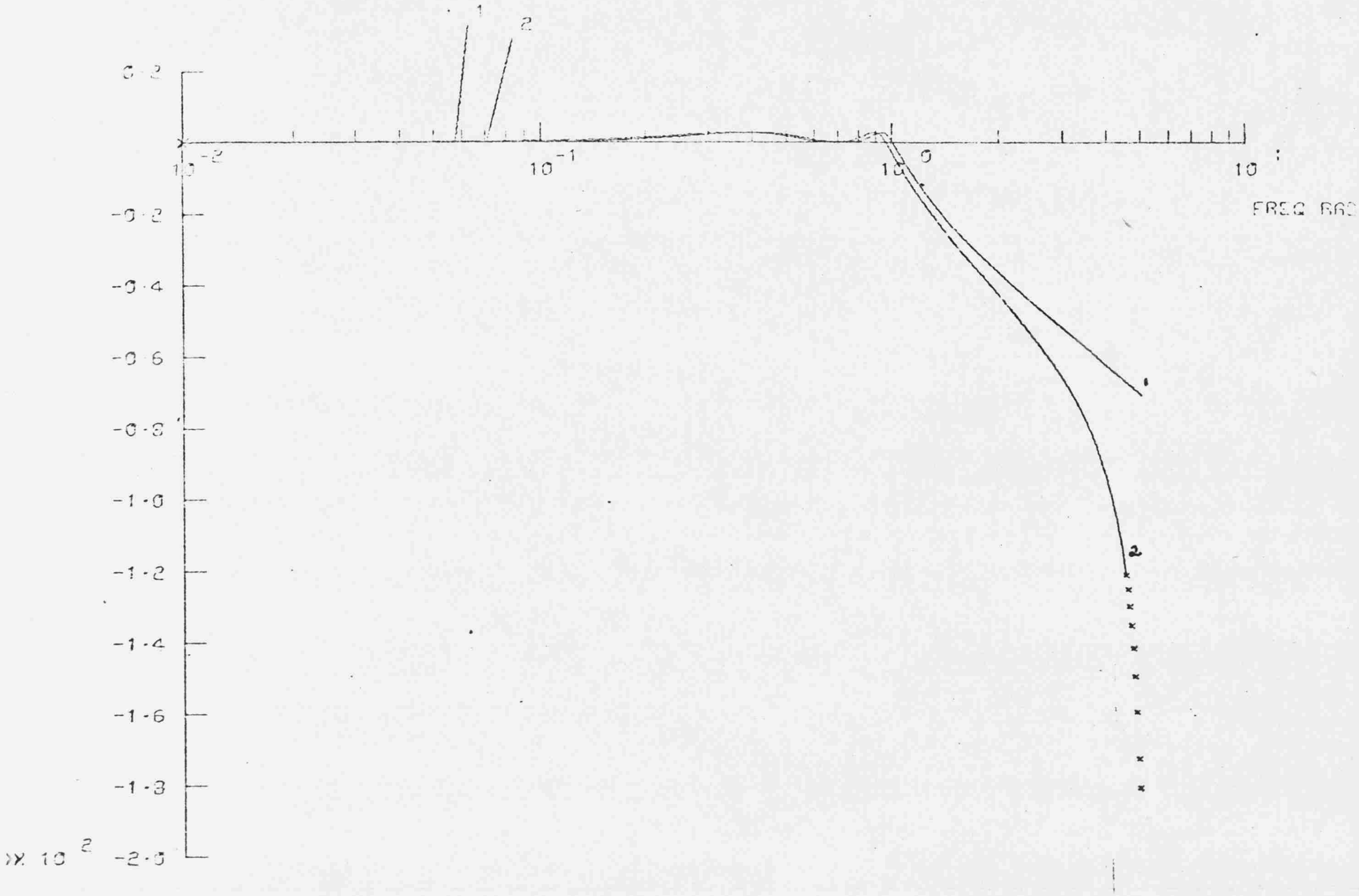
FILTER COMPARISON



PHASE 1-CONTINUOUS 2-DIGITAL

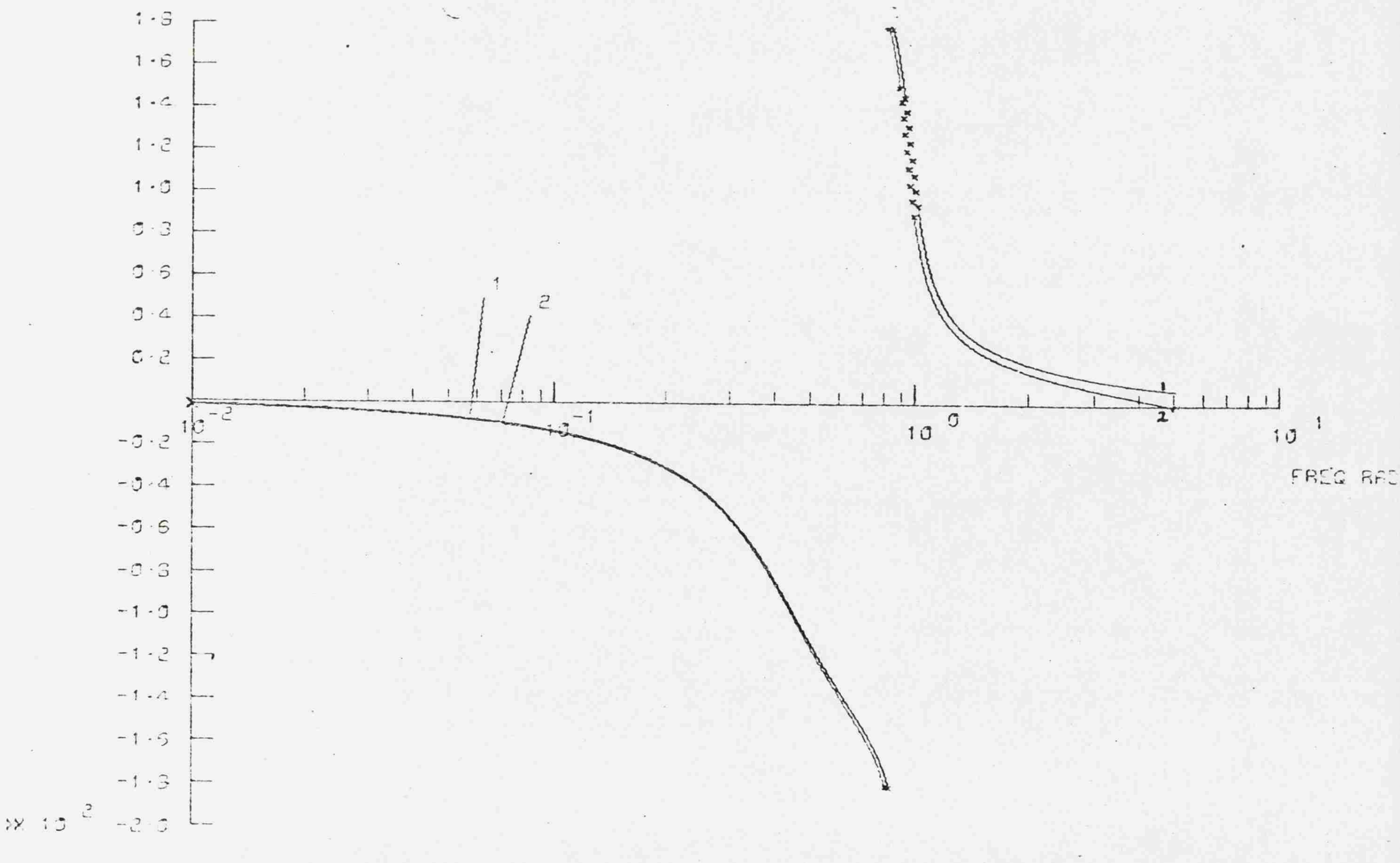
THE STANDARD Z TRANSFORM RESULT

FIGURE 2.1



DB GAIN 1-CONTINUOUS 2-DIGITAL

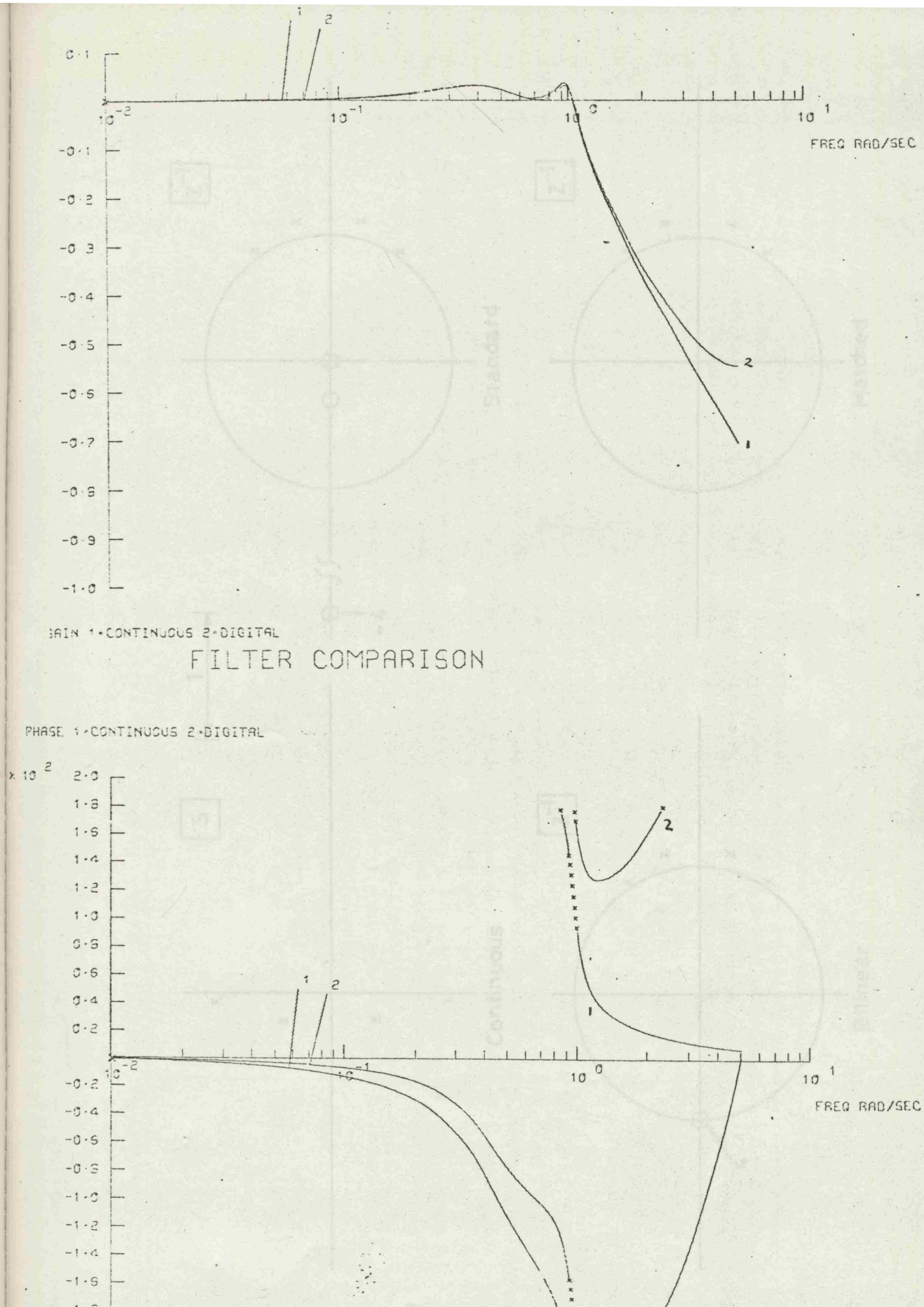
FILTER COMPARISON



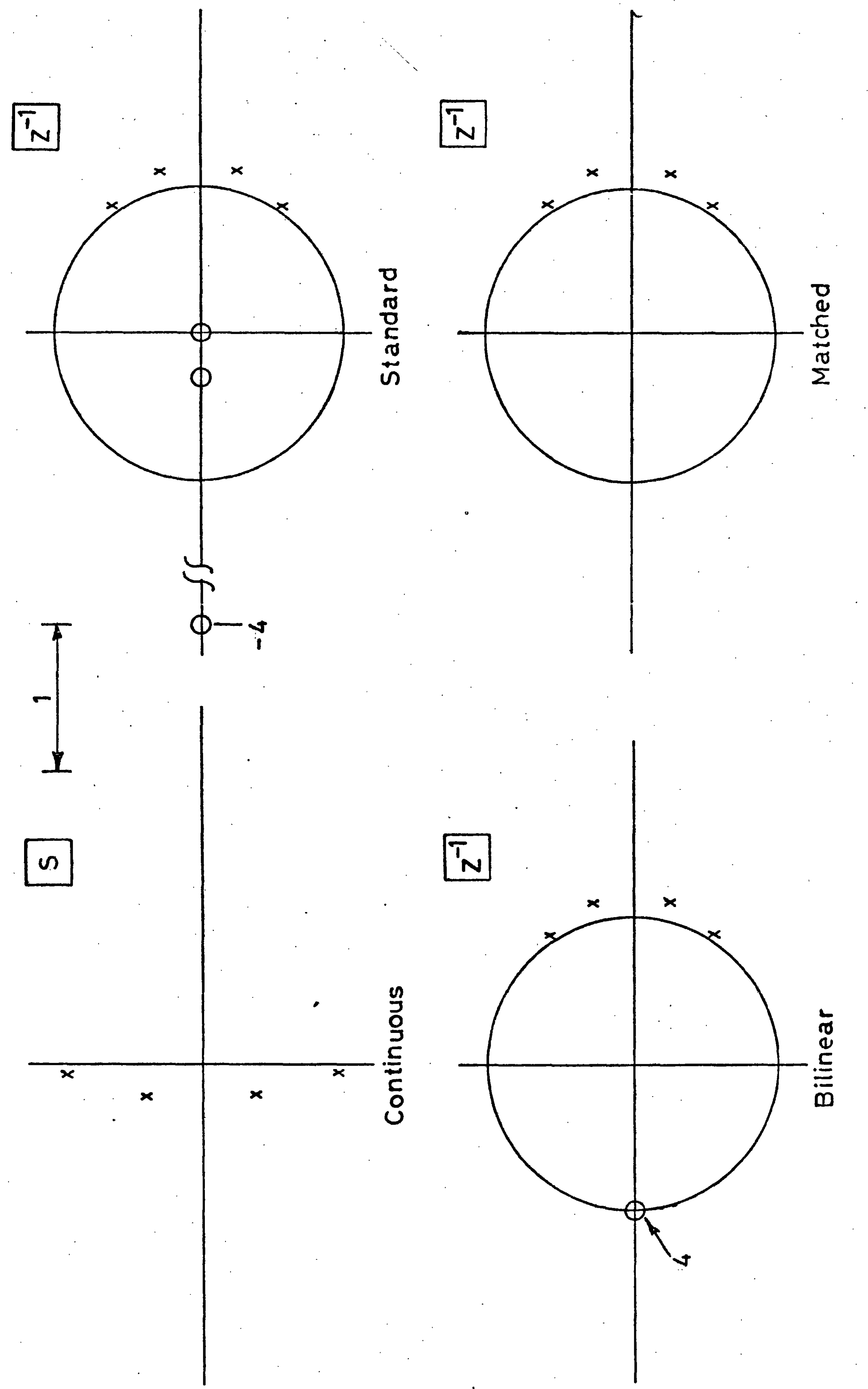
PHASE 1-CONTINUOUS 2 DIGITAL

THE BILINEAR Z TRANSFORM RESULT

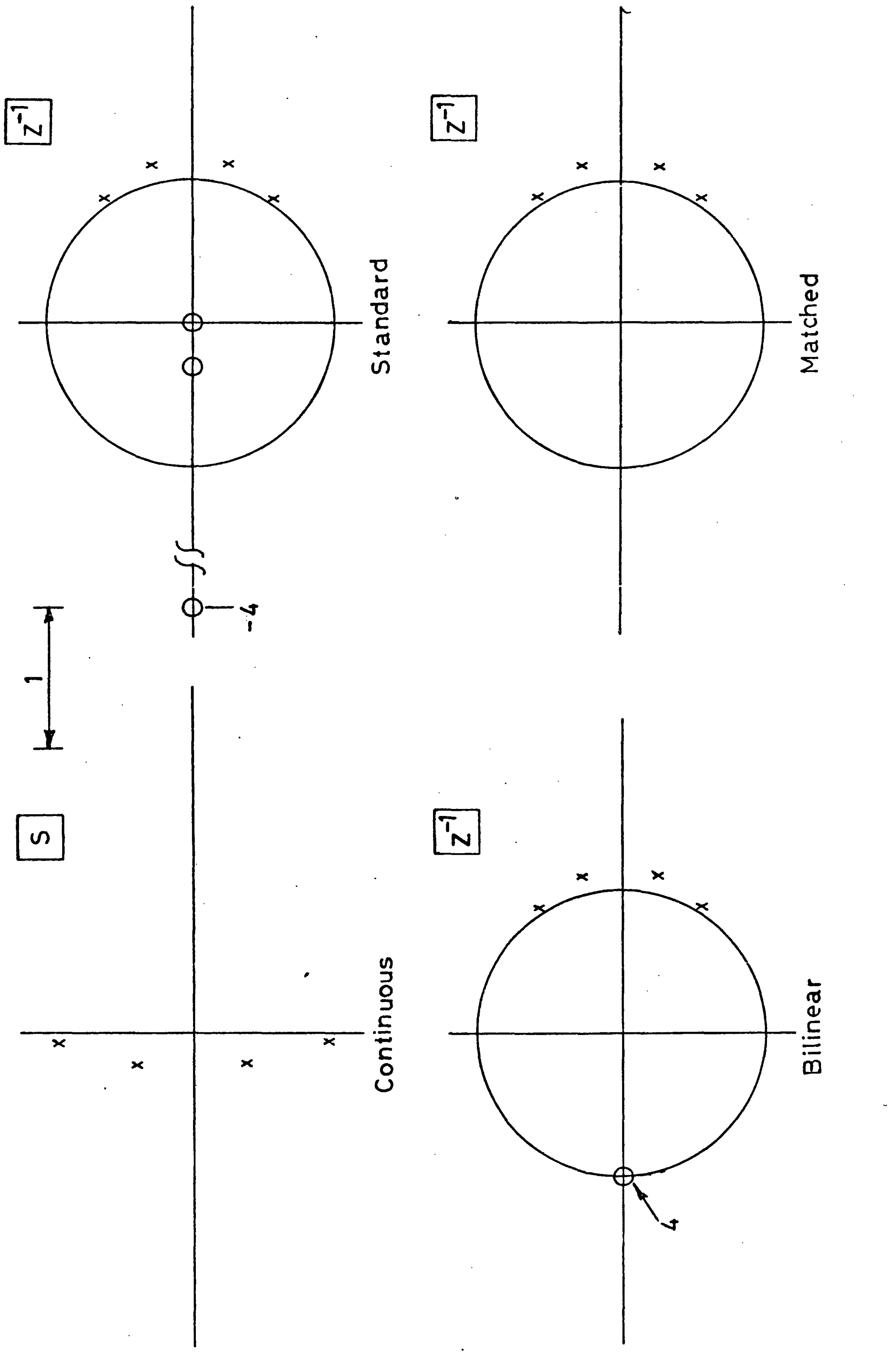
FIGURE 2.2



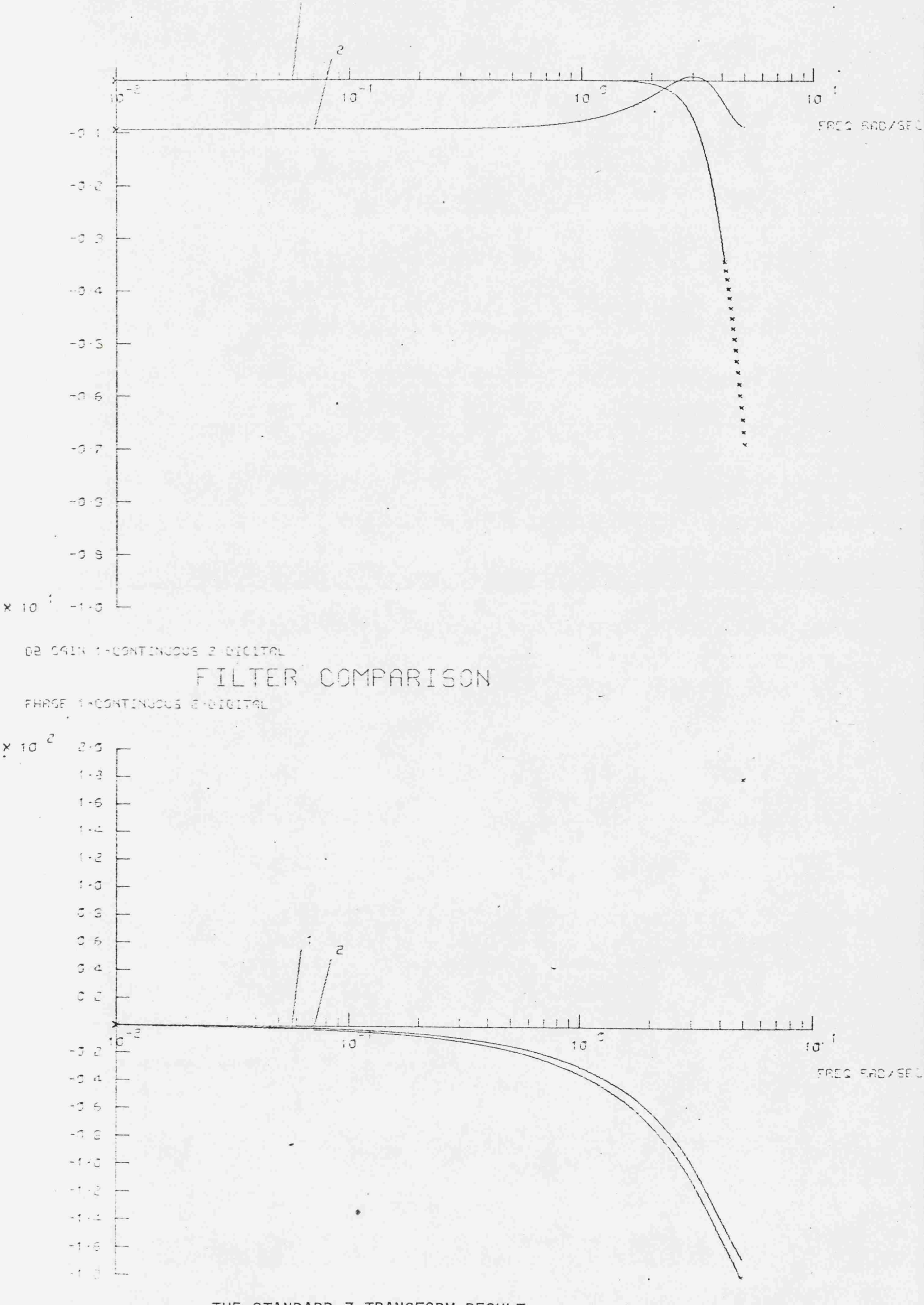
THE MATCHED Z TRANSFORM RESULT FIGURE 2.3



POLE-ZERO PLOTS FOR THE 4TH ORDER CHEBYSHEV FILTER FIGURE 2.4

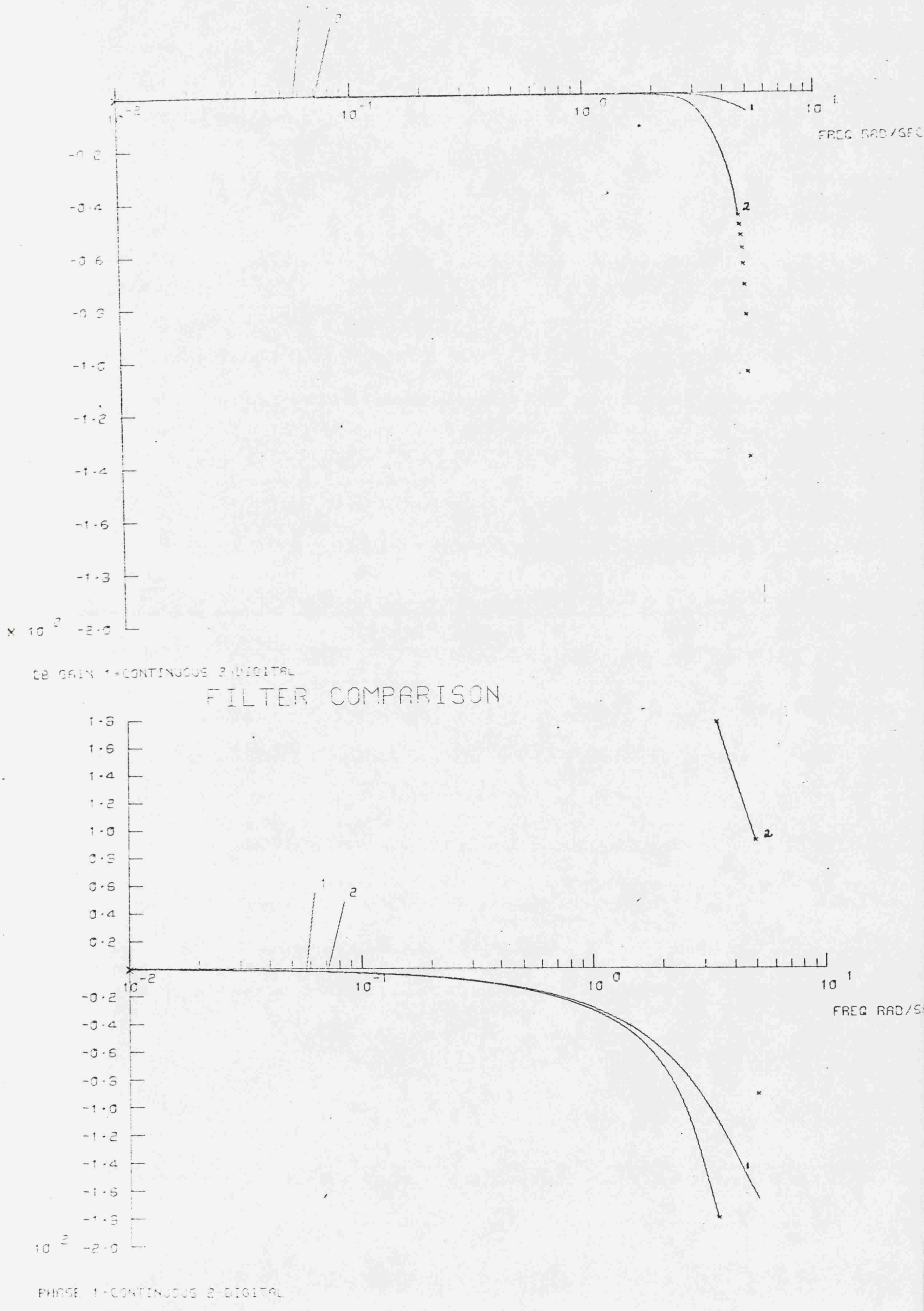


POLE-ZERO PLOTS FOR THE 4TH ORDER CHEBYSHEV FILTER FIGURE 2.4

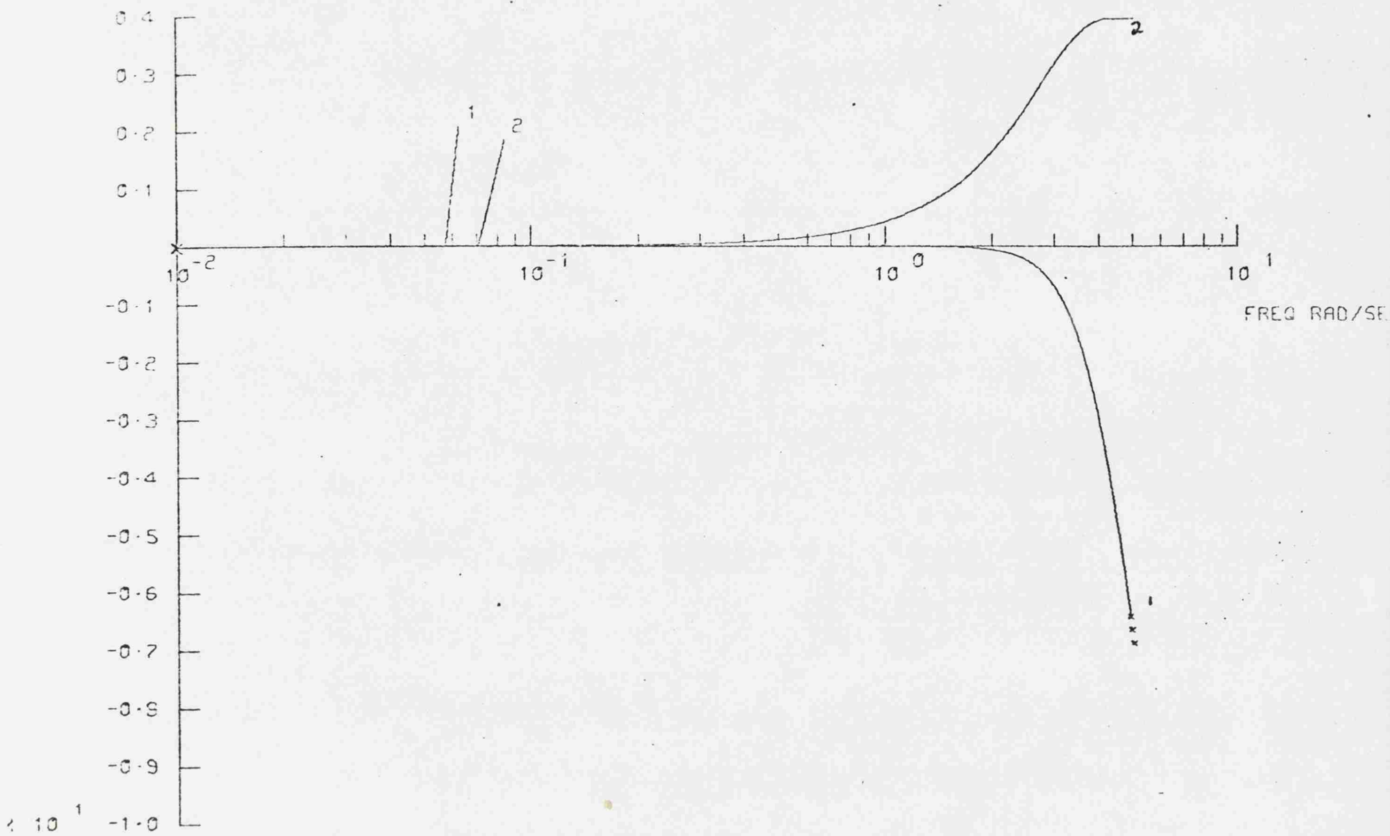


THE STANDARD Z TRANSFORM RESULT

FIGURE 2.5

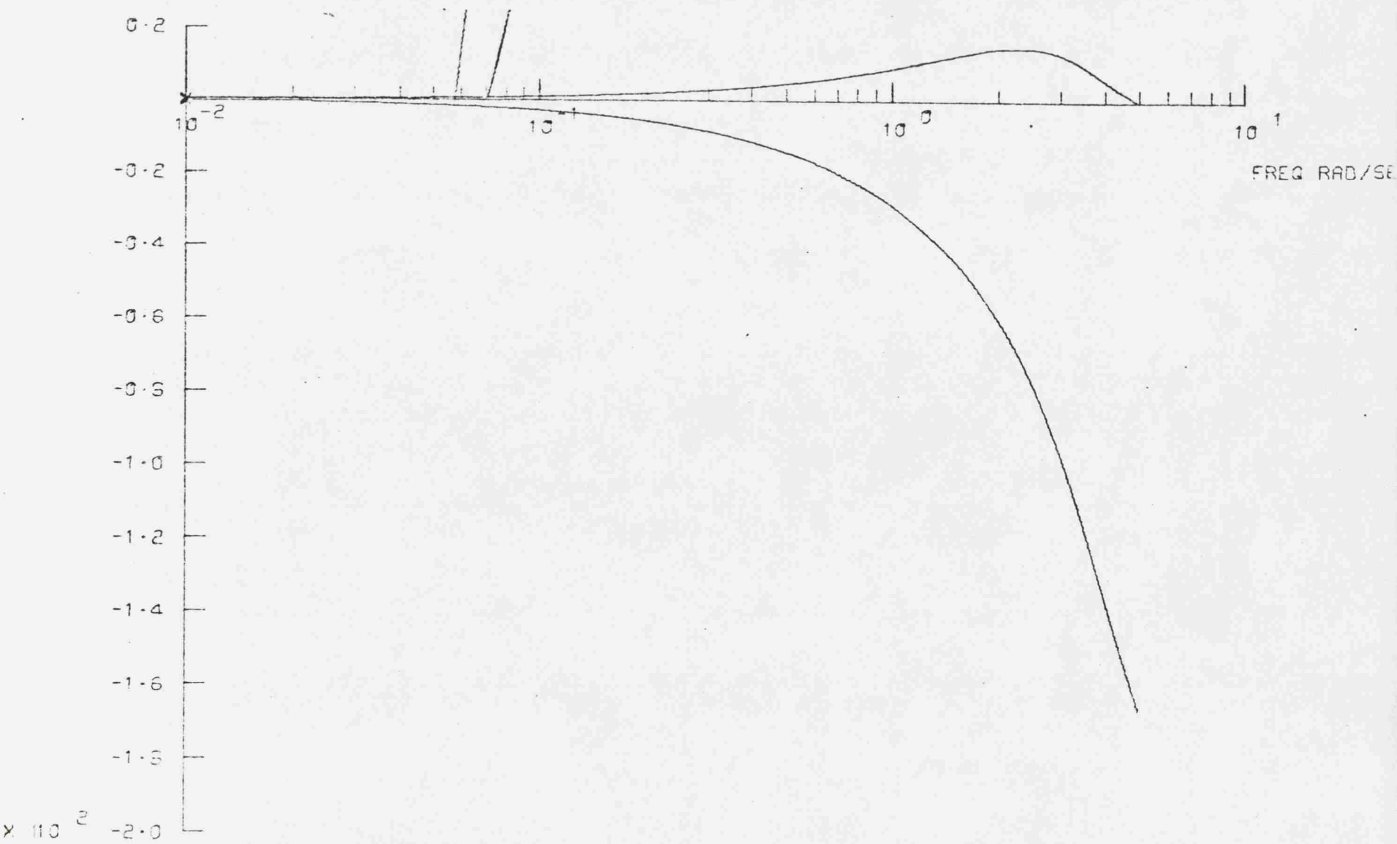


THE BILINEAR Z TRANSFORM RESULT FIGURE 2.6



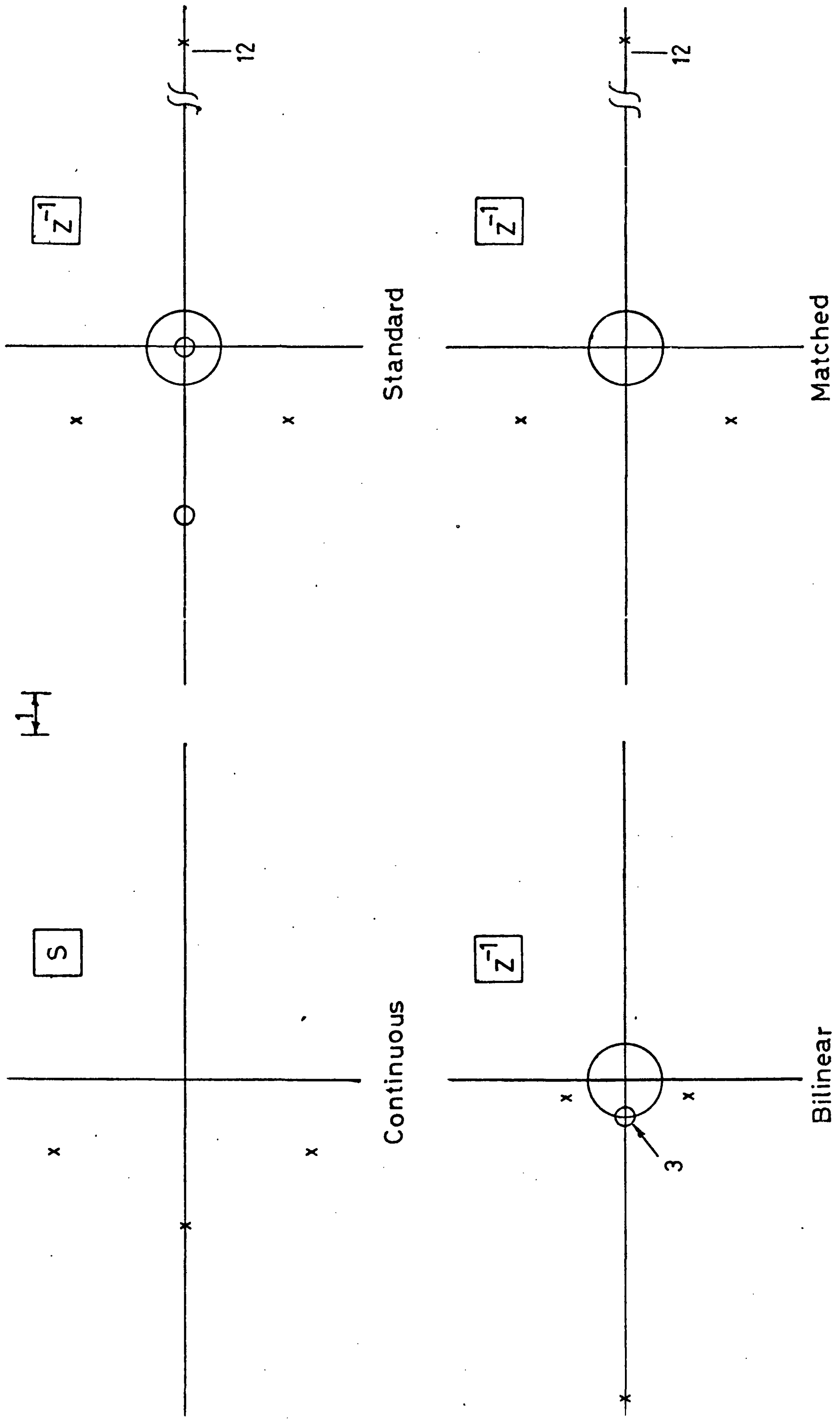
DB GAIN 1=CONTINUOUS 2 DIGITAL

FILTER COMPARISON

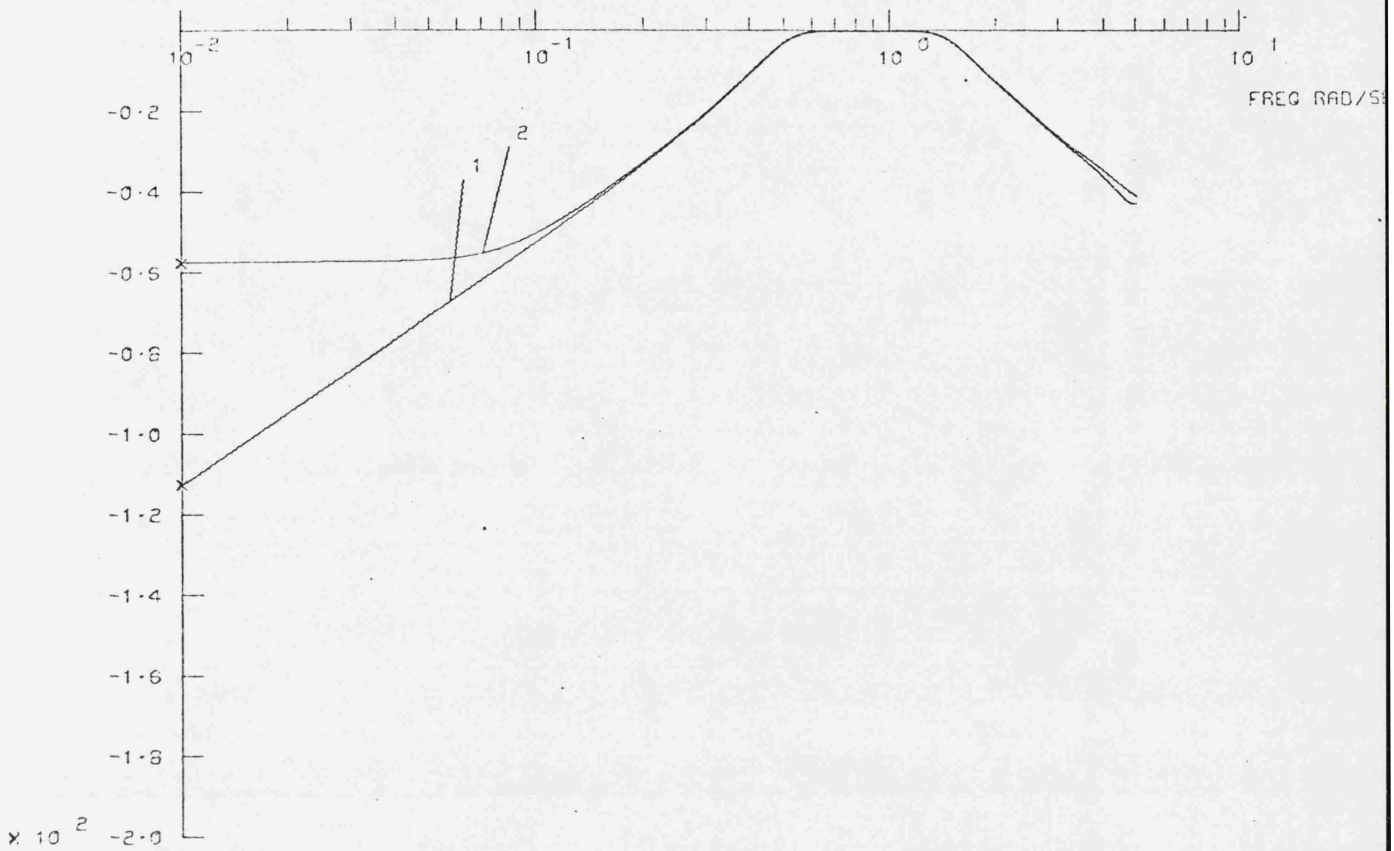


PHASE 1-CONTINUOUS 2 DIGITAL

THE MATCHED Z TRANSFORM RESULT FIGURE 2.7



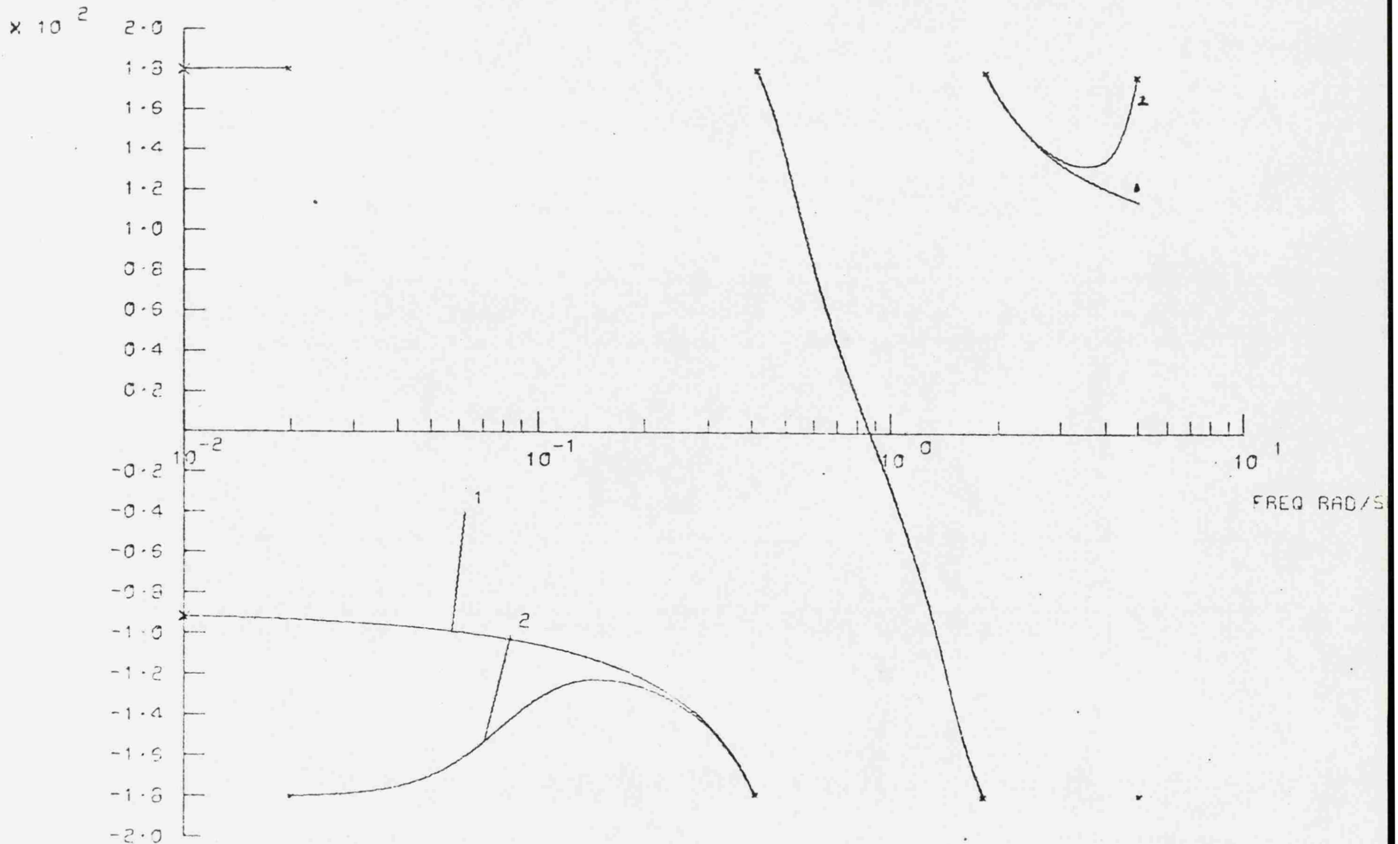
POLE-ZERO PLOTS FOR THE 3RD ORDER BUTTERWORTH FILTER FIGURE 2.8



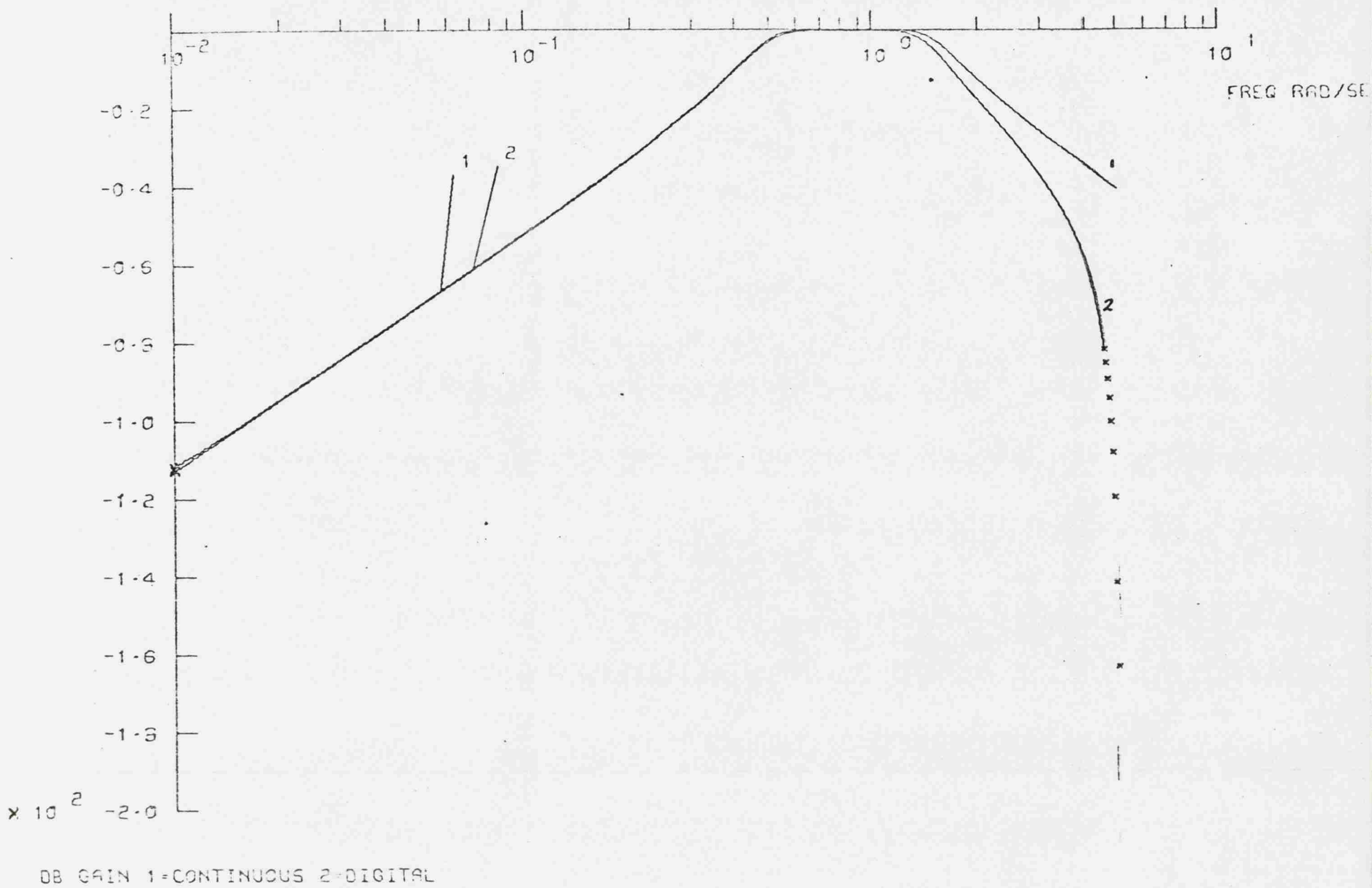
DB GAIN 1=CONTINUOUS 2-DIGITAL

FILTER COMPARISON

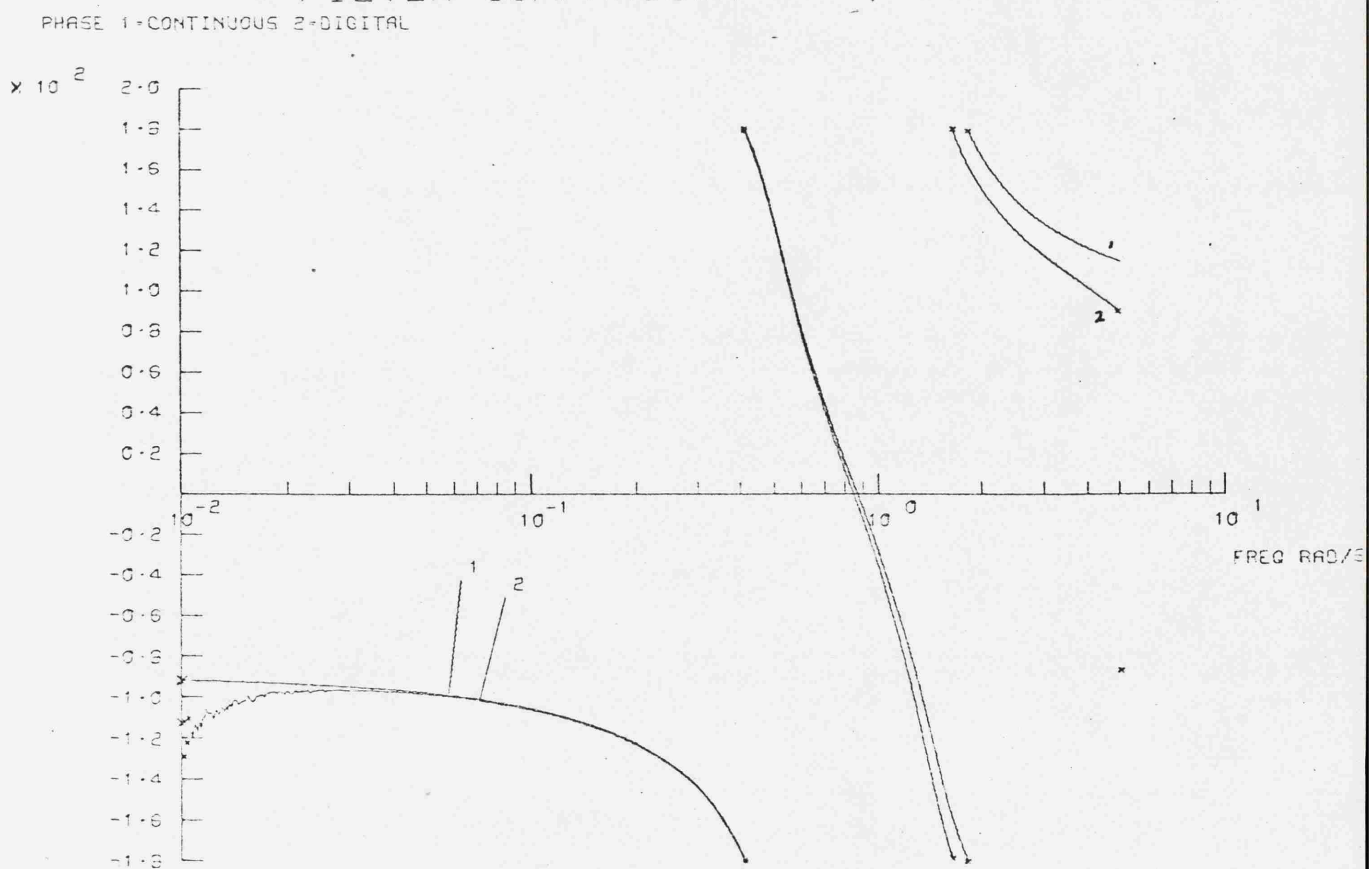
PHASE 1-CONTINUOUS 2-DIGITAL



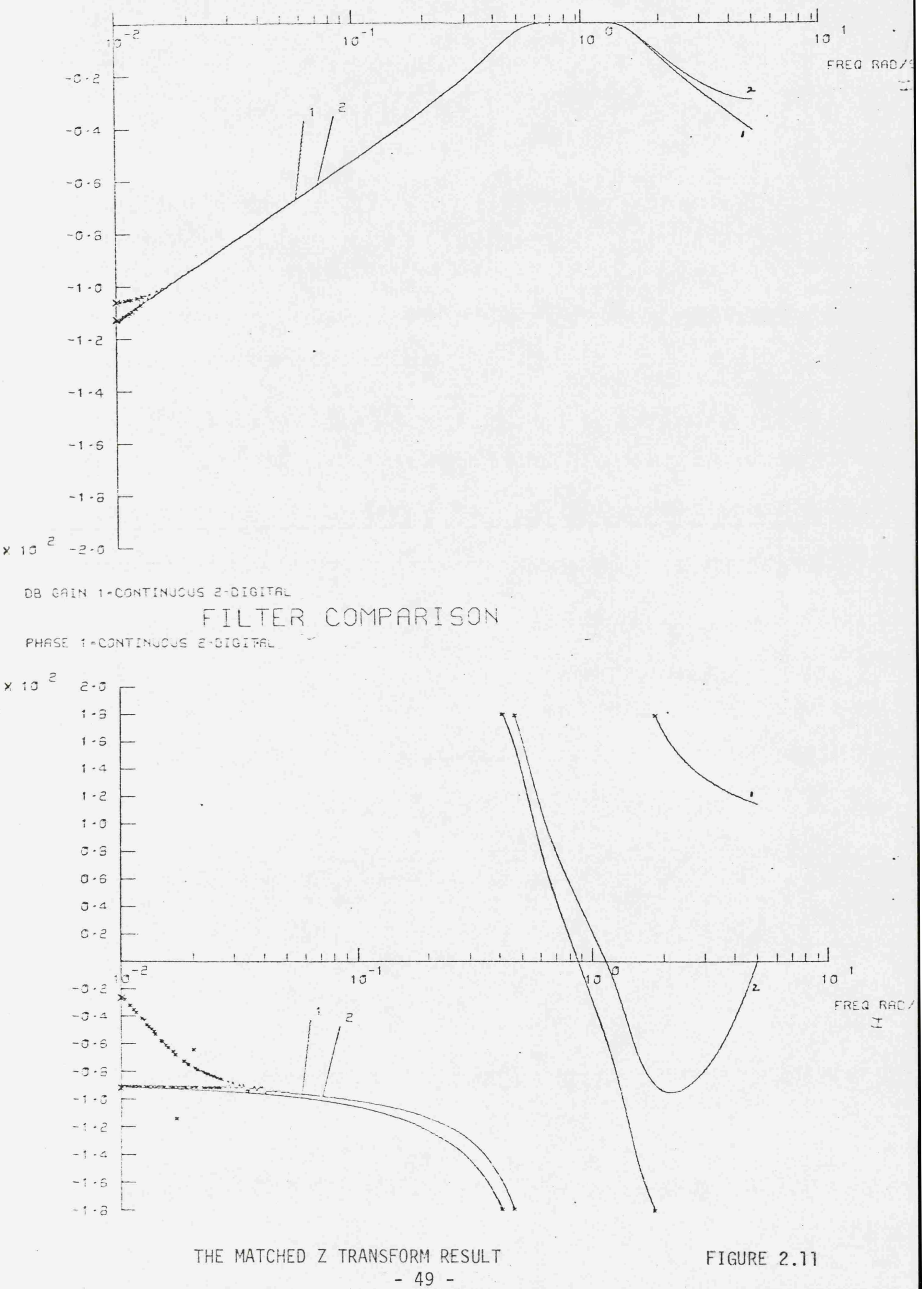
THE STANDARD Z TRANSFORM RESULT FIGURE 2.9



FILTER COMPARISON

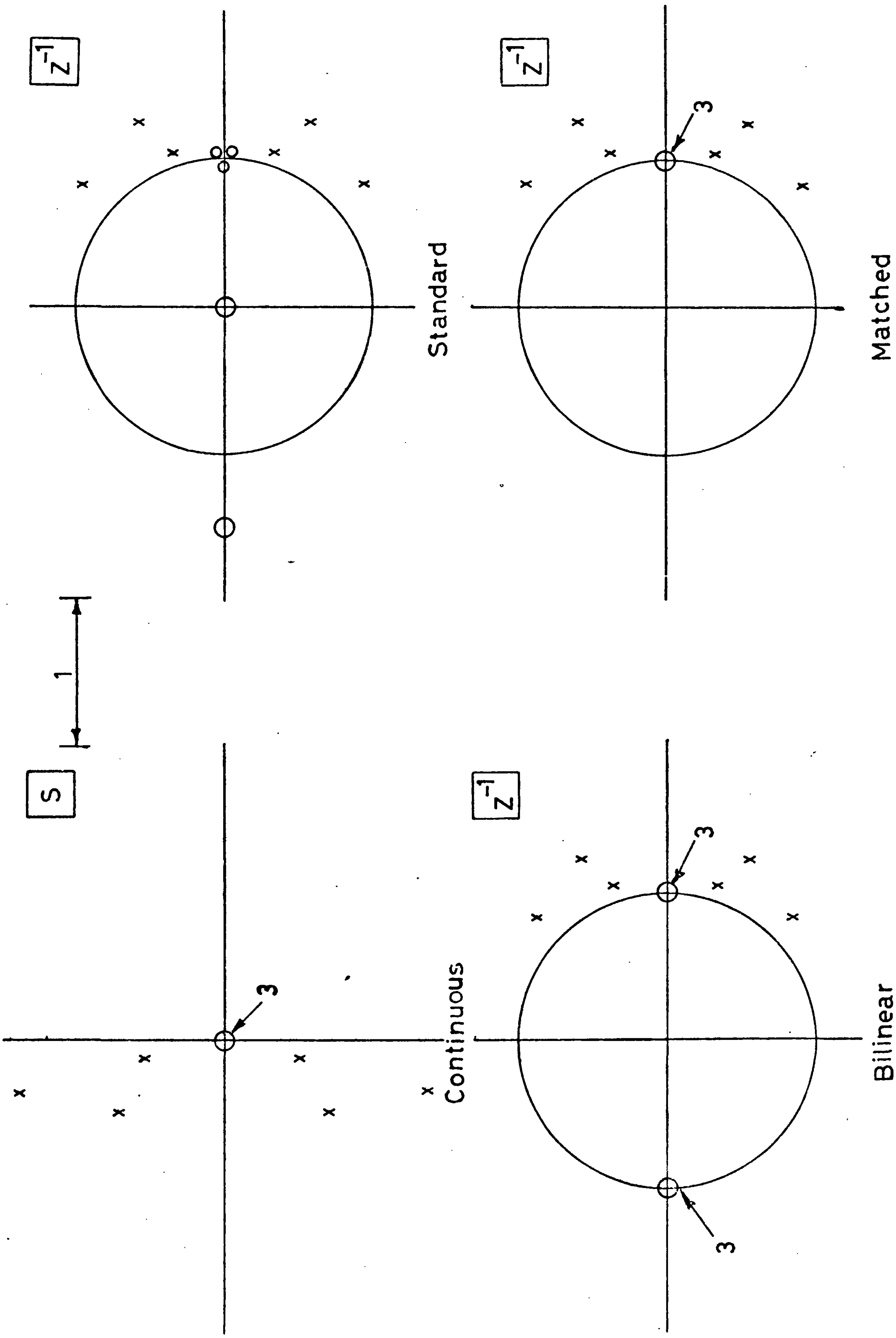


THE BILINEAR Z TRANSFORM RESULT FIGURE 2.10

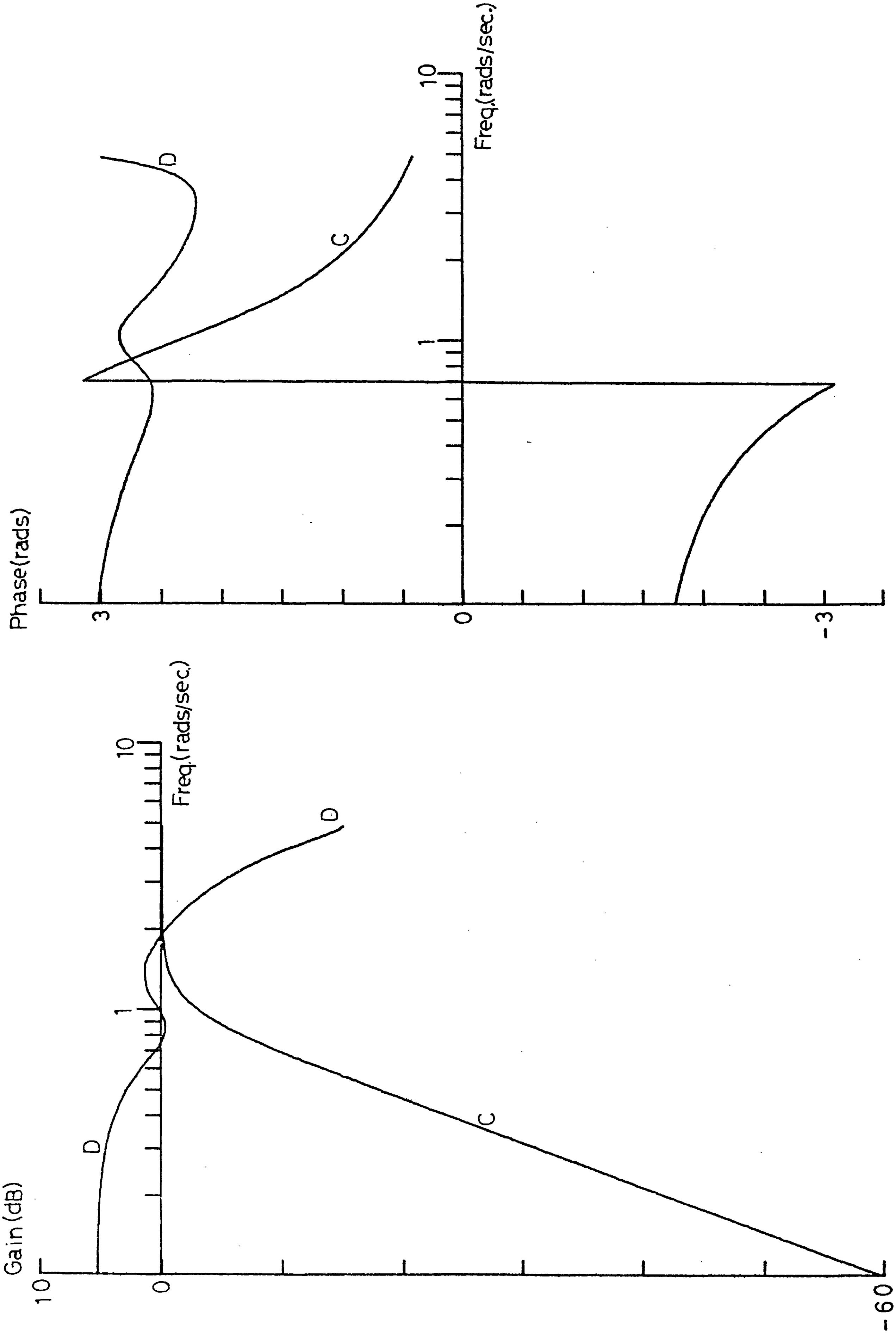


THE MATCHED Z TRANSFORM RESULT

FIGURE 2.11

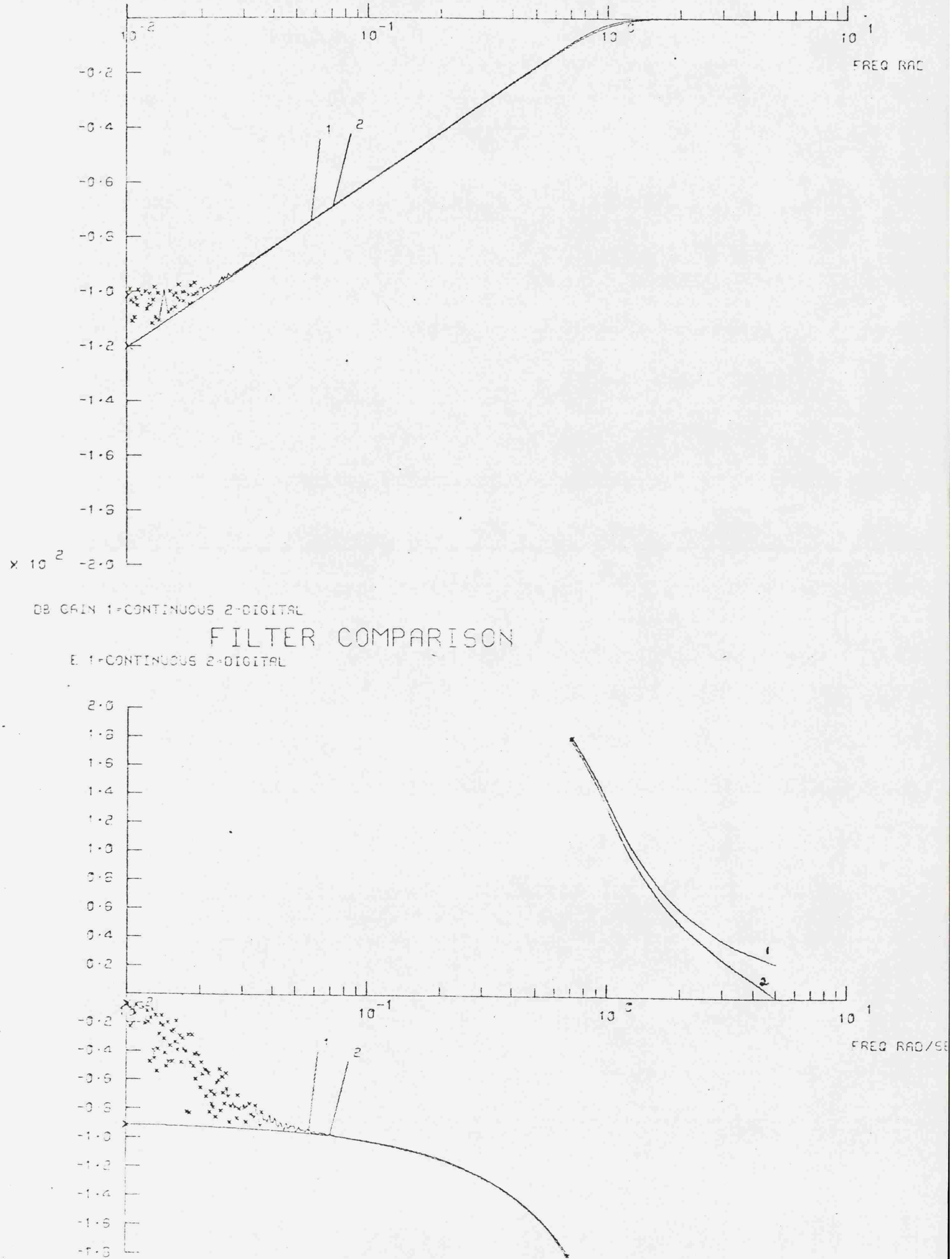


POLE-ZERO PLOTS FOR THE 6TH ORDER BAND-PASS FILTER FIGURE 2.12



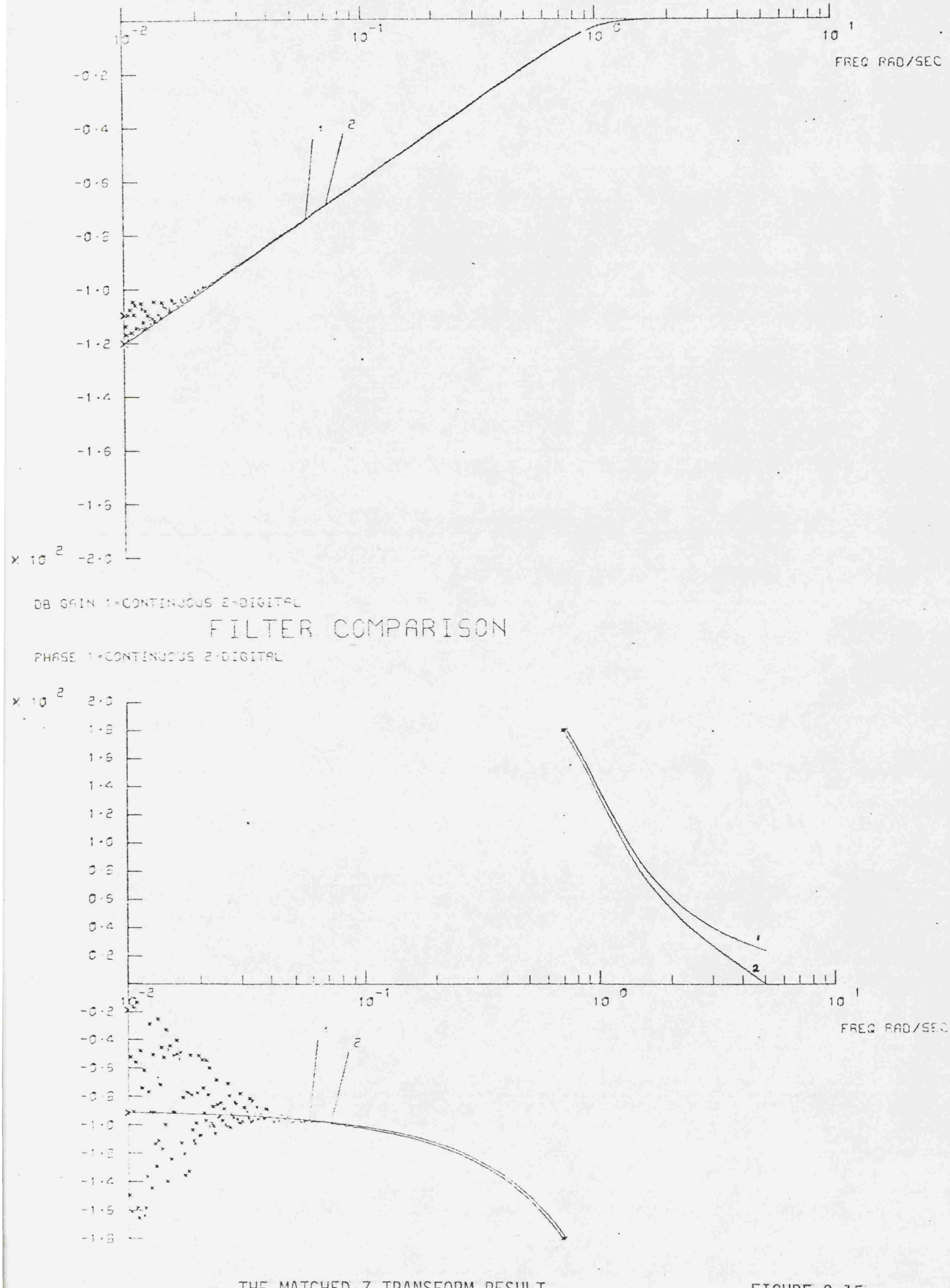
THE STANDARD Z TRANSFORM RESULT

FIGURE 2.13



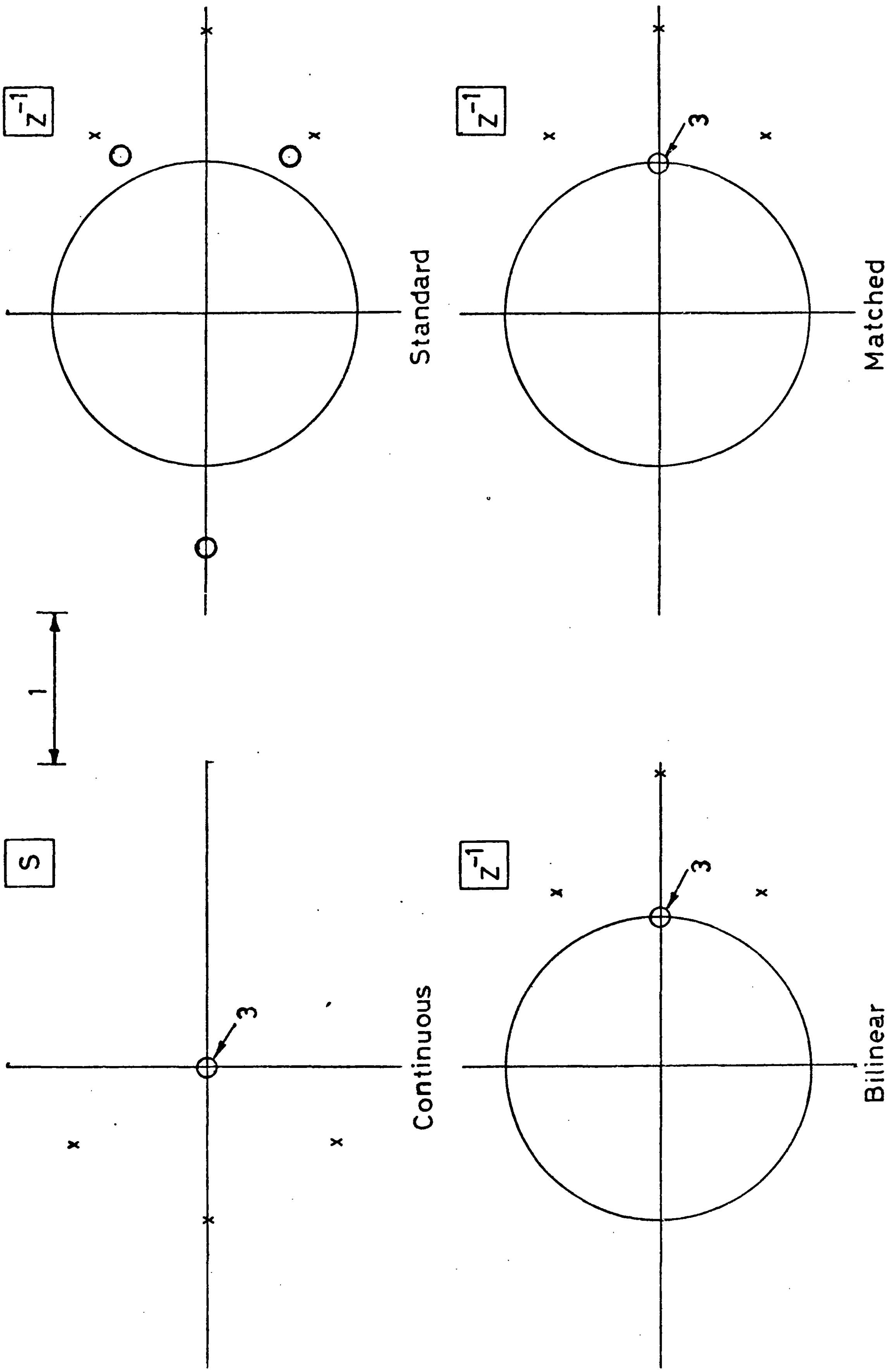
THE BILINEAR Z TRANSFORM RESULT

FIGURE 2.14



THE MATCHED Z TRANSFORM RESULT

FIGURE 2.15



POLE-ZERO PLOTS FOR THE BUTTERWORTH HIGH-PASS FILTER

FIGURE 2.16

3. THE SIMULATION PROBLEM

Section 2.1 illustrated that a measure of accuracy of simulation of frequency response can be obtained by careful choice of transformation method. The question must be asked as to what constitutes a good simulation, with respect to the design problems of digital filters. In Section 1.4 it was stated that the only real basis for accuracy is the comparison of frequency responses. To review, it was clear that for time economic digital simulation the sampling rate must be as low as possible as must the order of the digital simulation. Due to the preponderance of interest in practical digital filters, the literature shows no great interest in the comparison of digital and continuous frequency responses. Thus the question of accuracy in this context must be studied in a more general sense.

Consider a real, causal, recursive digital transfer function (with real coefficients a_i, b_k, c_l and d_p), and written in factored form:

$$H(Z^{-1}) = \frac{\prod_{i=1}^{m} (a_i + Z^{-1}) \prod_{k=1}^{n} (b_{1k} + b_{2k} Z^{-1} + Z^{-2})}{\prod_{\ell=1}^{u} (c_\ell + Z^{-1}) \prod_{p=1}^{v} (d_{1p} + d_{2p} Z^{-1} + Z^{-2})}, \qquad 3.1$$

where a_i and c_ℓ specify real roots

and b_k and d_p specify complex conjugate pairs of roots.

Further consider the situation at the Nyquist frequency (when $Z^{-1} = -1$). Then the value of $H(Z^{-1})$ can only be real or zero and not complex.

3.1 Phase Considerations

At $Z^{-1} = -1$:

(a) complex conjugate pairs of roots contribute $\pm 2\Pi$ radians phase shift to $\angle H(Z^{-1})$;

(b) non-zero real roots contribute 0 or $\pm\Pi$ radians phase shift to $\angle H(Z^{-1})$.

For a root placed at $Z^{-1} = -1$ the phase contribution is indeterminate. The equation of the root is:

$$A(Z^{-1}) = 1 + Z^{-1}, \tag{3.2}$$

the frequency response of which is:

$$G(\omega) = 1 + e^{-j\omega T}. \tag{3.3}$$

The phase contribution at the Nyquist frequency ϕ_n is given by:

$$\phi_n = \lim_{\omega \to \frac{\pi}{T}} \angle G(\omega), \tag{3.4}$$

now:

$$\lim_{\omega \to \frac{\pi}{T}} \angle G(\omega) = \lim_{\delta x \to 0} \tan^{-1} \left\{ \frac{-\sin(\pi - \delta x)}{1 + \cos(\pi - \delta x)} \right\}, \tag{3.5}$$

when $\delta x = 0$,

$$\angle G(\omega) = \tan^{-1} \frac{0}{0}. \tag{3.6}$$

Applying l'Hopital's rule:

$$\lim_{\omega \to \frac{\pi}{T}} \angle G(\omega) = \lim_{\delta x \to 0} \tan^{-1} \left\{ \frac{\cos(\pi - \delta x)}{\sin(\pi - \delta x)} \right\}, \tag{3.7}$$

$$= \tan^{-1}(-\infty), \tag{3.8}$$

$$= \pm\frac{\pi}{2}. \tag{3.9}$$

Therefore as the phase contributions of the various elements are additive:

$$\angle H(Z^{-1})\Big|_{Z^{-1}=-1} = \pm\frac{n\pi}{2} \qquad 3.10$$

where n is integer.

3.2 Modulus Considerations

(At $Z^{-1} = -1$)

Redefining Equation 3.1 in terms of frequency function factors:

$$H(\omega) = \frac{\prod_{i=1}^{I} P_i(\omega)}{\prod_{k=1}^{K} q_k(\omega)} \qquad 3.11$$

then:

$$|H(\omega)| = \frac{\prod_{i=1}^{I} |P_i(\omega)|}{\prod_{k=1}^{K} |q_k(\omega)|} \quad . \qquad 3.12$$

For convenience the above is re-written:

$$|H(\omega)| = \frac{\prod_{i=1}^{I} f_i(\omega)}{\prod_{k=1}^{K} g_k(\omega)} \quad . \qquad 3.13$$

Now consider, (defining $h'(\omega) = \frac{d}{d\omega} h(\omega)$)

$$\frac{d}{d\omega}|H(\omega)| = \left[\left\{\sum_{i=1}^{I}\left(\prod_{\ell=1}^{I}\frac{f_\ell(\omega)\ f_i'(\omega)}{f_i(\omega)}\right)\right\}\left\{\prod_{k=1}^{K} g_k(\omega)\right\}\right.$$

$$\left.-\left\{\sum_{n=1}^{K}\left(\prod_{m=1}^{K}\frac{g_m(\omega)\ g_n'(\omega)}{g_n(\omega)}\right)\right\}\left\{\prod_{p=1}^{I} f_p(\omega)\right\}\right] \Bigg/ \left\{\prod_{r=1}^{K} g_r(\omega)\right\}^2 , \qquad 3.14$$

$$= |H(\omega)| \left\{ \sum_{i=1}^{I} \frac{f_i'(\omega)}{f_i(\omega)} - \sum_{k=1}^{K} \frac{g_k'(\omega)}{g_k(\omega)} \right\} . \qquad 3.15$$

Now, for real roots:

$$r(\omega) = a + e^{-j\omega T} \qquad 3.16$$

$$\frac{d}{d\omega} |r(\omega)| = \frac{-aT \sin \omega T}{(a^2 + 1 + 2a \cos \omega T)^{\frac{1}{2}}} . \qquad 3.17$$

Therefore:

$$\frac{\frac{d}{d\omega} |r(\omega)|}{|r(\omega)|} = \frac{-a\ T \sin \omega T}{a^2 + 1 + 2a \cos \omega T} . \qquad 3.18$$

For complex conjugate pairs of roots:

$$r(\omega) = (a + jb + e^{-j\omega T})(a - jb + e^{-j\omega T}) , \qquad 3.19$$

$$|r(\omega)| = \{a^4+b^4+4a^2+1-2a^2b^2+4a(a^2-b^2+1)\cos \omega T + 2(a^2-b^2)\cos 2\ \omega T\}^{\frac{1}{2}} , \qquad 3.20$$

giving:

$$\frac{\frac{d}{d\omega} |r(\omega)|}{|r(\omega)|} = \frac{-2aT(a^2-b^2+1)\sin \omega T - 2T(a^2-b^2)\sin 2\omega T}{a^4+b^4+4a^2+1-2a^2b^2+4a(a^2-b^2+1)\cos \omega T + 2(a^2-b^2)\cos 2\omega T} \qquad 3.21$$

Now for $Z^{-1} = -1$ and considering non-zero roots only:

$$\left. \frac{\frac{d}{d\omega}|r(\omega)|}{|r(\omega)|} \right|_{\omega=\frac{\pi}{T}} = 0 \qquad 3.22$$

Consider the case:

$$r(\omega) = 1 + e^{-j\omega T} , \qquad 3.23$$

that is, a root at $Z^{-1} = -1$.

Then, from Equation 3.18:

$$\frac{\frac{d}{d\omega}|r(\omega)|}{|r(\omega)|} = \frac{0}{0} . \qquad 3.24$$

Applying l'Hopitals rule:

$$\lim_{\omega \to \frac{\pi}{T}} \frac{\frac{d}{d\omega}|r(\omega)|}{|r(\omega)|} = -\infty . \qquad 3.25$$

Therefore from Equations 3.15, 3.22 and 3.25:

$$\frac{d}{d\omega} |H(\omega)| = 0 \text{ or } \pm \infty. \qquad 3.26$$

3.3 Discussion

For a finite order, real, causal digital filter the condition of the frequency response at the Nyquist frequency is such that:

(a) the slope of the magnitude response is zero or $\pm\infty$;

(b) the phase shift is $\frac{n\pi}{2}$.

These conditions mean that exact simulation of a continuous frequency response is virtually impossible. This conclusion arises because the original frequency response would have to comply with the above conditions. Such a compliance is very unlikely. The above conditions are necessary but not sufficient to state that exact simulation is impossible. However the conditions are sufficiently prohibitive for further investigations to concentrate upon the degree of closeness obtained from approximate solutions. The effect of the conditions can be seen clearly in the examples (Figures 2.1 to 2.15) presented in connection with the discussions of Section 2.1.

(Note: in some practical situations the continuous filter frequency response response can be very close to the above conditions and in these cases the approximation problems will not be great).

4. THE ZERO INSERT METHOD

Section 2.1 demonstrated the effects of using Z transformation methods. Each of the transformation methods was shown to introduce some frequency response discrepancy. The pole-zero plots of the resultant transfer functions demonstrated that the frequency response differences are mainly due to the placing of the Z^{-1} plane zeros. The Standard Z Transform can give superior performance to the Matched Z Transform where infinite s plane zeros exist. When infinite s plane zeros are present the Standard Z Transform produces finite Z^{-1} plane zeros, whilst the Matched Z Transform maintains infinite zeros in the Z^{-1} plane. However where finite 's' plane zeros exist the Matched Z transform can give superior performance. It seems, therefore, that if the Matched Z Transform is used and some account can be taken of infinite s plane zeros, then an accurate frequency response simulation may result. Thus the object of this section of the work is to study the use of the Matched Z Transform together with inserted Z^{-1} plane zeros.

Section 2.1 also shows that the Bilinear Z Transform deals with infinite s plane zeros by placing zeros at $Z^{-1} = -1$. Golden (Ref.24) studied transformation methods, and considered the use of zeros at $Z^{-1} = -1$ in addition to the Matched Z Transform. This idea produces results similar to those of the Bilinear Z Transform with pole-zero positions which are not warped with respect to the frequency variable. Golden's work is interesting in that it attempts to utilise extra zeros in the Z^{-1} plane. However the work here reported will consider more varied placings for the extra zeros in the Z^{-1} plane.

4.1 Frequency Response Effects of Z^{-1} Plane Zeros

Consider the equation of a zero in the Z^{-1} plane:

$$G(Z^{-1}) = k(r\, e^{j\theta} + Z^{-1}) \,. \qquad 4.1$$

Giving:

$$|G(Z^{-1})|\Big|_{Z^{-1} = e^{-j\omega T}} = k\{r^2 + 1 + 2r \cos(\omega T + \theta)\}^{\frac{1}{2}} \qquad 4.2$$

$$\angle G(Z^{-1})\Big|_{Z^{-1} = e^{-j\omega T}} = \tan^{-1}\left\{\frac{r \sin\theta - \sin\omega T}{r \cos\theta + \cos\omega T}\right\} \qquad 4.3$$

Giving a group delay:

$$D(\omega) = \frac{-T\{1 + r\cos(\omega T + \theta)\}}{r^2 + 2\, r \cos\,(\omega T + \theta) + 1} \qquad 4.4$$

Analysis of Equations 4.3 and 4.4 gives an interesting result. For a given value of r let:

$$\angle G(Z^{-1})\Big|_{Z^{-1} = e^{-j\omega T}} = -f(\omega), \qquad 4.5$$

then for a zero placed at $Z^{-1} = \frac{1}{r} e^{j\theta}$,

$$\angle G(Z^{-1})\Big|_{Z^{-1} = e^{-j\omega T}} = -\omega T + f(\omega). \qquad 4.6$$

In addition, from Equation 4.2, the gain relationships for r and $\frac{1}{r}$ are similar, and become identical if a suitable adjustment is made to the overall gain k. This result can be important in that it allows slight flexibility in the choice of phase response for a given gain relationship. The relevance of these points will become clear later.

4.1.1 Real Zeros

For a zero placed on the real axis in the Z^{-1} plane Equation 4.1 can be written:

$$G(Z^{-1}) = k(b + Z^{-1}). \qquad 4.7$$

The values of b and k may be found by calculation if the gain of the zero is known at two points in the frequency range. For illustrative purposes let:

$$G(Z^{-1})\Big|_{Z^{-1}=1} = 1 \qquad 4.8$$

and

$$G(Z^{-1})\Big|_{Z^{-1}=-1} = g, \qquad 4.9$$

then:

$$k = \frac{1-g}{2} \quad \text{and} \quad b = \frac{1+g}{1-g} . \qquad 4.10$$

Writing Equation 4.2 in terms of g gives:

$$|G(Z^{-1})|\Big|_{Z^{-1}=e^{-j\omega T}} = \frac{1}{\sqrt{2}}\left\{1 + g^2 + (1 - g^2)\cos\omega T\right\}^{\frac{1}{2}} . \qquad 4.11$$

This relationship is independent of the sign of g and if the sign of g is reversed the effect is to move the zero from $Z^{-1} = -b$ to $Z^{-1} = -\frac{1}{b}$, this result being due to the effects discussed with respect to Equations 4.5 and 4.6. Another observation that can be made is that if:

$b > 0$ then $|g| < 1$ and if,

$b < 0$ then $|g| > 1$.

Figure 4.1 shows curves of $|G(Z^{-1})|$ versus frequency for various values of $|g|$.

The overall effects of real zeros can be summarised in tabular form (Table 4.1). In the table the gain of the zero at zero frequency ($Z^{-1} = 1$) is assumed to be unity, and the value of $|b|$ is assumed to be greater than one. It is to be noted that in the table $f(\omega)$ is positive for ω positive.

4.1.2 Complex Conjugate Pairs of Zeros

The equation of a complex conjugate zero pair can be written:

$$H(Z^{-1}) = k(r\,e^{j\theta} + Z^{-1})(r\,e^{-j\theta} + Z^{-1}). \qquad 4.12$$

The gain magnitude of such an equation shows a minimum at a frequency closely related to the angle of the zero positions. The phase response of a complex conjugate pair of zeros has cumulative relationship of two responses of the type discussed with respect to Equations 4.5 and 4.6. Therefore if:

$$\angle H(Z^{-1})\Big|_{Z^{-1} = e^{-j\omega T}} = -f(\omega) \qquad 4.13$$

then for zeros whose magnitude value is $1/r$:

$$\angle H(Z^{-1})\Big|_{Z^{-1} = e^{-j\omega T}} = -2\omega T + f(\omega). \qquad 4.14$$

When this result is considered with respect to the actual values of phase involved, it is seen that:

$$\angle H(Z^{-1})\Big|_{Z^{-1}=1} = 0 \text{ radians,} \qquad 4.15$$

and

$$\angle H(Z^{-1})\Big|_{Z^{-1}=-1} = 0 \text{ radians (if } r>1\text{),} \qquad 4.16$$

$$= -2\pi \text{ radians (if } r<1\text{).} \qquad 4.17$$

4.2 Considerations With Respect to the Matched Z Transform

The phase response of a continuous transfer function is asymptotic to a multiple of $\pi/2$ radians as the frequency approaches infinity. The phase response of a digital filter is equal to a multiple of $\pi/2$ radians at the Nyquist frequency. The intention of this work is to produce digital responses which approximate continuous responses. This modelling is only possible over the Nyquist range, thus the relevant extent of the continuous response is truncated at the Nyquist frequency. At the proposed Nyquist frequency a given continuous response may well not be approaching its phase asymptote. Therefore the digital model Nyquist frequency phase must be determined from the actual phase requirement at this point.

In the case of the Matched Z Transform the Nyquist frequency phase will be $p\pi/2$ radians less than the asymptotic phase of the continuous prototype, where p infinite zeros exist (see examples in Section 2). This shortfall is due to the infinite zeros in the 's' plane becoming infinite zeros in the Z^{-1} plane when using the Matched Z Transform.

Section 4.1 showed that the phase response of inserted zeros can be adjusted, to some extent, without affecting the magnitude response. The extent of the possible adjustment, for a given amplitude response, is limited to considerations of the Nyquist frequency phase shift. For illustrative purposes, suppose that it has been decided to insert a zero on the negative real axis in the Z^{-1} plane to achieve a particular magnitude requirement. Then various arrangements of zeros could be considered.

(i) using a zero at $Z^{-1} = -b$ $(b>1)$, in this case the phase response will be $-f(\omega)$ where $f(\omega)$ is positive for positive ω and is zero at the Nyquist frequency;

(ii) using a zero at $Z^{-1} = -1/b$, in this case the phase response will be $-\omega T + f(\omega)$ and will be $-\pi$ radians at the Nyquist frequency;

(iii) using a zero at $Z^{-1} = -b$ and an extra zero at $Z^{-1} = 0$ (which will not affect the magnitude response), then the phase response will be $-\omega T - f(\omega)$.

Thus for a particular phase response requirement some control is available even though the magnitude response is fixed.

Therefore, for a given Matched Z Transform design let 'n' be the number of $-\pi/2$ radian phase shifts required at the Nyquist frequency. From Section 3 it can be seen that if 'n' is odd a zero is required at $Z^{-1} = -1$. Such a zero may be undesirable, in which case the phase will have to be adjusted to the nearest increment of $-\pi$ radians. Let 'm' be the number of zeros to be placed inside the unit circle in the Z^{-1} plane, then:

for n even $m = n/2$

for n odd $m = (n-1)/2$, or if no zero is to be placed at $Z^{-1} = -1$ $m = (n\pm1)/2$. 4.20

From the consideration of Matched Z Transform designs in Chapter 2 and the discussion in this section it is clear that the magnitude-frequency response is increasingly too high as the Nyquist frequency is approached. Zeros placed in the negative real Z^{-1} axis have a magnitude response minimum at the Nyquist frequency. Therefore useful results are most likely to occur when additional zeros are placed on the negative real axis in the Z^{-1} plane.

4.3 The Zero Insertion

For a system with a large number of s plane infinite zeros the problems in correctly inserting zeros in the Z^{-1} plane are complicated. A large number of zero configurations are possible and to arrive at a good configuration in an economic way is difficult. The precise positions of the inserted zeros will depend upon the criteria used to assess the quality of the response. Due to these complexity problems it is intended to consider the relatively simple case of inserting two real zeros in an attempt to show what can be accomplished by the zero insertion method.

4.3.1 The Effect of Inserting a Pair of Real Zeros

Consider the equation of two real zeros:

$$G(Z^{-1}) = k_1(b + Z^{-1})\, k_2\,(c + Z^{-1}), \qquad 4.21$$

and let $b < 1$ and $c > 1$ (one zero inside and one outside the unit circle), also let:

$$G(Z^{-1}) = 1 \quad \text{for} \quad Z^{-1} = 1$$

and $$G(Z^{-1}) = g \quad \text{for} \quad Z^{-1} = -1 \,. \qquad 4.22$$

Therefore if:

$$k_1(b - 1) = g_1$$

and

$$k_2(c - 1) = g_2$$

then

$$g = g_1 g_2 \,. \qquad 4.23$$

As $b < 1$ the phase shift at the Nyquist frequency is $-\pi$ radians and g_1 is negative. Consider the case:

$$g_1 = -g_2 = \sqrt{g} \qquad 4.24$$

then

$$b = \frac{1}{c} \,. \qquad 4.25$$

For this case the phase response:

$$\phi(\omega) = -\omega T, \qquad 4.26$$

(from Section 4.1.1 and Table 4.1) which is a linear phase.

From the considerations of the phase response of single zeros, and now considering the case where $|g_1| \neq |g_2|$ then, if:

$$|g_1| > |g_2| \quad \text{then} \quad \phi(\omega) < -\omega T \qquad 4.27$$

and if $$|g_2| > |g_1| \quad \text{then} \quad \phi(\omega) > -\omega T \,. \qquad 4.28$$

This result is arrived at from the fact that, if one zero is allowed to have more effect, then the phase deviation ($f(\omega)$ in Table 4.1) of that zero will be greater thus producing the effect of Equations 4.27 and 4.28. It is true however that, if the gain magnitude values are allowed to change, the overall magnitude response will also change. The question of what criteria are used to assess the quality of the resultant response will thus dictate what is done with regard to the above considerations. The values of the zeros can be determined from three points on the frequency response, required magnitude or phase values can be used at these points to evaluate the precise zero positions. In the overwhelming majority of cases one of these points will be $Z^{-1} = 1$, and the gain of the zeros at this point will be unity (due to the nature of the Matched Z Transform result). In a typical case the magnitude of $G(Z^{-1})$ could be determined at $Z^{-1} = 1$ and $Z^{-1} = -1$ and the precise zero positions calculated from the deviation of the required phase response from the linear phase $-\omega T$.

4.3.2 An Example Using Two Real Zeros

Consider the example of a fourth order, low-pass, Chebyshev filter. The frequency response comparison for the Matched Z Transform can be seen in Figure 2.3. Figure 2.1 (the Standard Z Transform comparison) shows that a good simulation is possible in this case. For this example there are four infinite 's' plane zeros. Let 'Θ' be the continuous filter phase shift at the Nyquist frequency. Let 'n' be the number of $\pi/2$ radian phase shifts required, such that the digital filter phase shift is adjusted to nearest increment of $\pi/2$ radians to the continuous filter phase shift at the Nyquist frequency. Then:

$$n = \frac{2\Theta}{\pi} , \qquad 4.29$$

rounded to the nearest integer. In this example $n = 4$.

Now as:

$$\Theta - \frac{n\pi}{2} < 0, \qquad 4.30$$

the required phase response:

$$\phi(\omega) < -\frac{n}{2}\ \omega T, \qquad 4.31$$

and therefore zeros placed inside the unit circle should be allowed to have more effect. Equation 4.20 applied to this case shows that two zeros need to be placed inside the unit circle. In this example two zeros are to be used to modify the frequency response. These two zeros may be placed inside or outside the Z^{-1} plane unit circle. However two zeros must be placed inside the unit circle if the phase response requirements are to be met. For the purposes of demonstration the gain effective zeros will be placed such that one lies inside and one outside the unit circle in the Z^{-1} plane. To meet the phase response requirement an additional zero will be placed at $Z^{-1} = 0$. This action will have no effect upon the magnitude of the gain. The other two zero positions can now be found by calculation from the value of the required response at three points. The calculation in this case is performed for unity gain at zero frequency, -14dB gain at the Nyquist frequency, and phase equalisation at 80% of the Nyquist frequency. For these conditions zeros are placed at $Z^{-1} = -0.419$ and -2.94, (a typical calculation of this sort will be shown later). These zero positions plus the zero at $Z^{-1} = 0$ and the Matched Z Transform result give the digital filter whose frequency response is compared to the continuous case in Figure 4.2.

4.3.3 Choosing the Zeros

The problems involved in placing the zeros in the example discussed in Section 4.3.2 were quite simple. The original error was not serious and therefore simple calculation involving only two zeros was sufficient. However, even in this simple example it was clear that a certain amount of experience of digital frequency responses was needed and the result was effectively found from an 'informed guess'. Such methods are not suitable for application in a simulation system which is to be of general usage. This is particularly true where the error of the original (Matched) frequency response is more serious, and where a larger number of zeros are to be used. The response error is:

$$e(\omega) = H(s) - G(Z^{-1})\Big|_{\substack{s = j\omega \\ Z^{-1} = e^{-j\omega T}}} \qquad 4.32$$

From Chapter 3 it is clear that $e(\omega)$ cannot be realised with a finite number of zeros. Thus the requirement is that the inserted zeros must approximate $e(\omega)$ as closely as possible. In stating this the question again arises as to, 'what is a measure of closeness?' For particular applications this 'measure' could be obtained by various means. In the work reported here no particular error minimisation has been used, the approach being rather heuristic. Later work in connection with other approaches to simulation filter design use particular error minimisation methods. These methods could have application in the zero insertion area, though they have not been specifically applied by the author. The approach to the zero insertion method has been more on the lines of demonstrating what can be achieved by the use of digital simulation filters. Thus the only other method used for finding the positions of inserted zeros was a simple root searching method using the error function $e(\omega)$. This method gave a first approximation to the zero positions which were then adjusted by interactive operation using a computer. The root searching method used was that of a first order approximation to the Taylor series (when applied to polynomials this method is known as a Newton-Raphson iteration method). Consider the Taylor series:

$$f(a+h) = f(a) + h\,f'(a) + \frac{h^2}{2!}\,f''(a) + \ldots \text{ etc.,} \qquad 4.33$$

where $$f'(a) = \frac{d}{da}\,f(a)\;,\; f''(a) = \frac{d^2 f(a)}{da^2}\;,\text{ etc.} \qquad 4.34$$

The first order approximation to Equation 4.33 is thus:

$$f(a + h) = f(a) + h\,f'(a). \qquad 4.35$$

If now it is assumed that:

$$f(a + h) = 0, \qquad 4.36$$

then 'h' can be calculated and a new value of 'a' found and this value is again used in Equation 4.35 for the next iteration. This iterative method will eventually converge on a value of 'a' where f(a) is zero. When a zero is found it is eliminated from the function and further zeros are searched for. In this way zeros can be found. In the case of $e(\omega)$ the zero positions found only form an approximation to the required zero insertion positions but the positions can give a good starting point for further adjustment.

4.4 Further Examples

In the graphs used to demonstrate the examples the curves are identified by the following notation:

C - continuous,

D - digital,

D_s - Standard Z Transform,

D_m - Matched Z Transform,

D_z - Matched Z Transform with added zeros at $Z^{-1} = -1$.

All of the following examples have been designed using a sampling frequency of 10 rads/sec.

(a) Figures 4.3, 4.4, 4.5 and 4.6 show the results for a 6th order, low pass, Butterworth filter of cut-off frequency 3.33 rads/sec. Figure 4.3 shows the Standard and Matched Z transform results, plus the response for the Matched Z transform with zeros placed at $Z^{-1} = -1$, as suggested by Golden (Ref. 24). Figure 4.4 shows the associated pole-zero diagrams.

(i) Figure 4.5, using zeros placed at $Z^{-1} = 0, -0.1225, -1.468$ and -4.973. These zero positions were found by the use of the root searching method as described in Section 4.3.

(ii) Figure 4.6, using zeros placed at $Z^{-1} = 0, -0.25, -1.4$, and -7.5. In this case the zero positions were found by interactive adjustment, starting with the positions found by the root-searching method.

(b) Figures 4.7, 4.8 and 4.9 show the results for a 4th order, band-pass filter with a centre frequency of 3 rads/sec.

In this case the continuous transfer function is:

$$H(s) = \frac{0.2s^2}{s^4 + 0.2s^3 + 18s^2 + 1.8 \quad + 81} \cdot \qquad 4.37$$

Figure 4.7 shows the transformation designs, as in the previous example, compared with the original response and Figure 4.8, the pole-zero diagrams. In this case, if the order of the digital filter is to be kept the same as the continuous prototype, only two zeros can be inserted. The positions of the zeros were found by simple calculation to be required at $Z^{-1} = -0.326$ and -3.845. Figure 4.9 shows the frequency response comparison for the resultant digital filter. The calculations are shown below.

The equation of two Z^{-1} plane zeros is:

$$G(Z^{-1}) = a + bZ^{-1} + cZ^{-2}. \qquad 4.38$$

At $\omega = 0$, and for unity gain:

$$a + b + c = 1. \qquad 4.39$$

At the Nyquist frequency:

$$a - b + c = -0.2985, \qquad 4.40$$

for the required gain and $-\pi$ radians phase shift.

Therefore $\qquad b = 0.64927. \qquad 4.41$

The phase shift of $G(Z^{-1})$ is given by:

$$\Theta(\omega) = \tan^{-1}\left\{\frac{-b \sin \omega T - c \sin 2\omega T}{a + b \cos \omega T + c \cos \omega T}\right\}. \qquad 4.42$$

The required phase shift at 4 rads/sec is 2.45 rads.

Using Equations 4.39, 4.41 and 4.42 gives:

$$a = 0.1951 \quad \text{and} \quad c = 0.1557.$$

These values of a, b and c give roots at:

$$Z^{-1} = -3.845 \quad \text{and} \quad -0.3259. \qquad 4.43$$

4.5 Discussion

The responses shown in the examples indicate the type of performance that can be expected from the zero insert method. The final responses show improvement over the results obtained simply by using the transformation methods. The schemes used to find the zero positions are rather limited and are intended merely to show what can be achieved. The limitations are that:

(i) the calculation method provides a fit only at specified points and thus does not necessarily give a good reduction of overall error;

(ii) the root searching method only provides a guide to the placing of zeros;

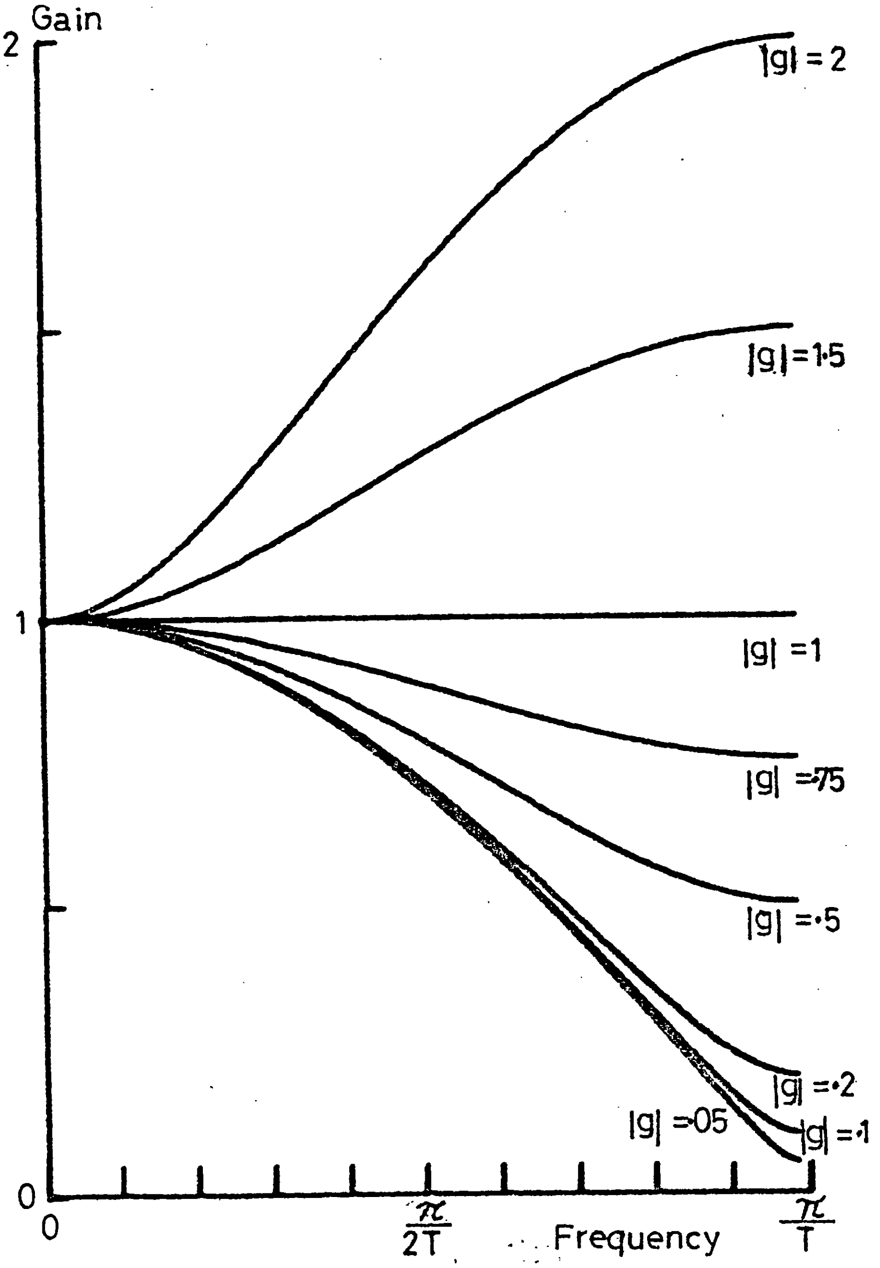
(iii) the interactive adjustment gives good results but is not really suitable from the point of view of generalised simulation usage. A method that would probably be more generally applicable would be a polynomial fitting method based upon a least square or a mini-max error criterion (a typical method is used in later sections for similar fitting problems).

Generally it can be said that the zero insert method shows promise as a method for designing digital simulation filters. The method has the advantage that it makes use of the good basis for design provided by the transformation methods. However the method is only applicable to the design of filters which use an 's' plane transfer function to specify the required device.

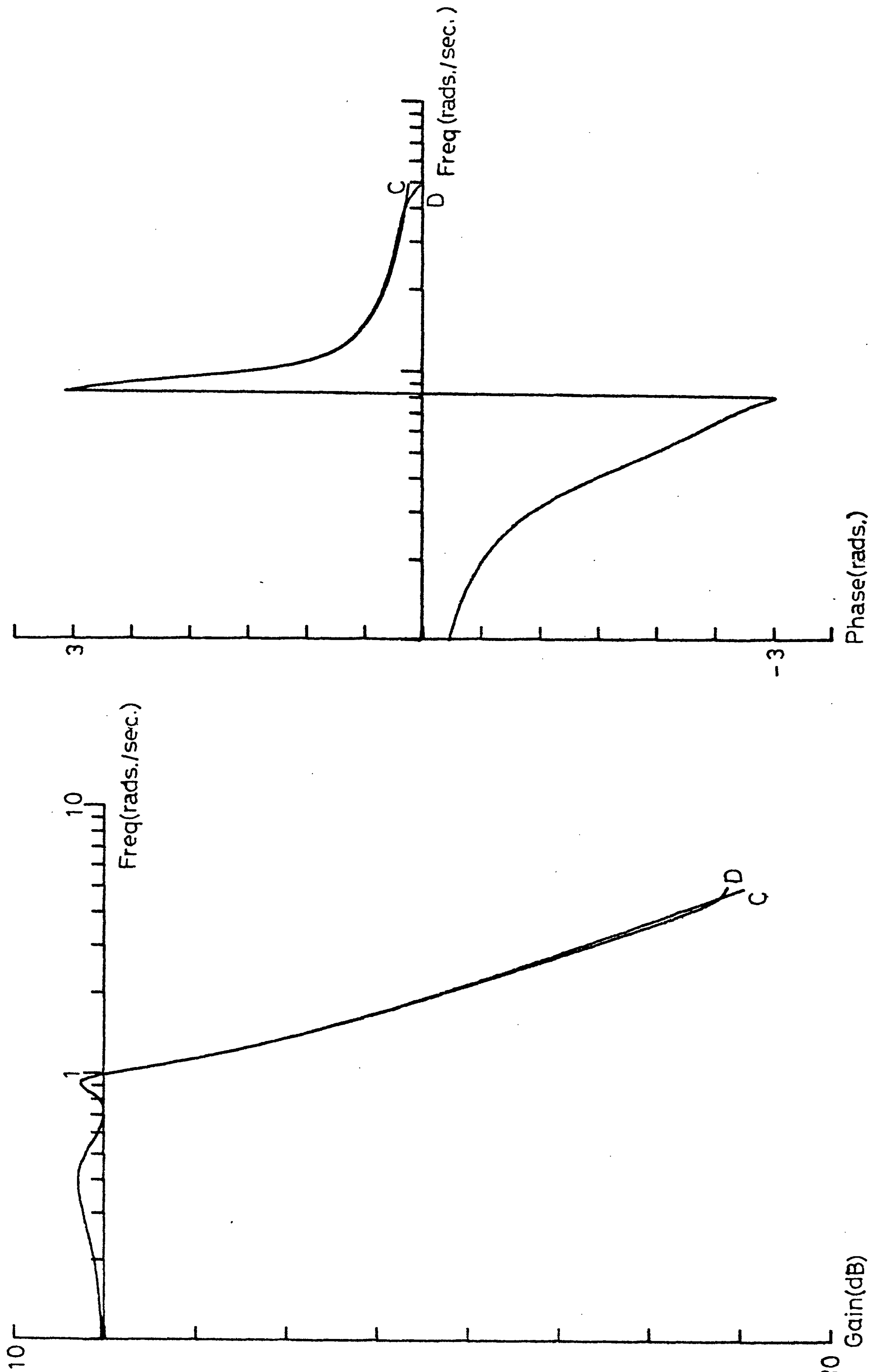
The study of the effect of placing zeros in the Z^{-1} plane gives an interesting insight into the relationships of digital filters. It must be noted that in the examples demonstrated and in other tests performed by the author, no use has been made of complex conjugate zeros. The examples and tests performed have not been sufficient to state that complex conjugate zeros will be of no use in response correction. The nature of the frequency response error of the Matched Z Transform, however, does indicate that real zeros are more likely to be useful. Further discussion of the method and a comparison with other simulation filter design methods can be found in Chapter 7.

Zero Position	Gain at $Z^{-1} = -1$	Phase Relationship	Phase at $Z^{-1} = 1$	Phase at $Z^{-1} = -1$
$Z^{-1} = -b$	g	$-f(\omega)$	0	0
$Z^{-1} = -\frac{1}{b}$	g	$-\omega T + f(\omega)$	0	$-\pi$
$Z^{-1} = \frac{1}{b}$	$\frac{1}{g}$	$-\omega T - f(\omega)$	0	$-\pi$
$Z^{-1} = b$	$\frac{1}{g}$	$-\pi + f(\omega)$	$-\pi$	$-\pi$
$Z^{-1} = -1$	0	$-\omega T/2$(linear)	0	$-\pi/2$
$Z^{-1} = 0$	1 (all pass)	$-\omega T$ (linear)	0	$-\pi$
$Z^{-1} = 1$	-	$-\frac{\pi}{2} - \frac{\omega T}{2}$	$-\frac{\pi}{2}$	0

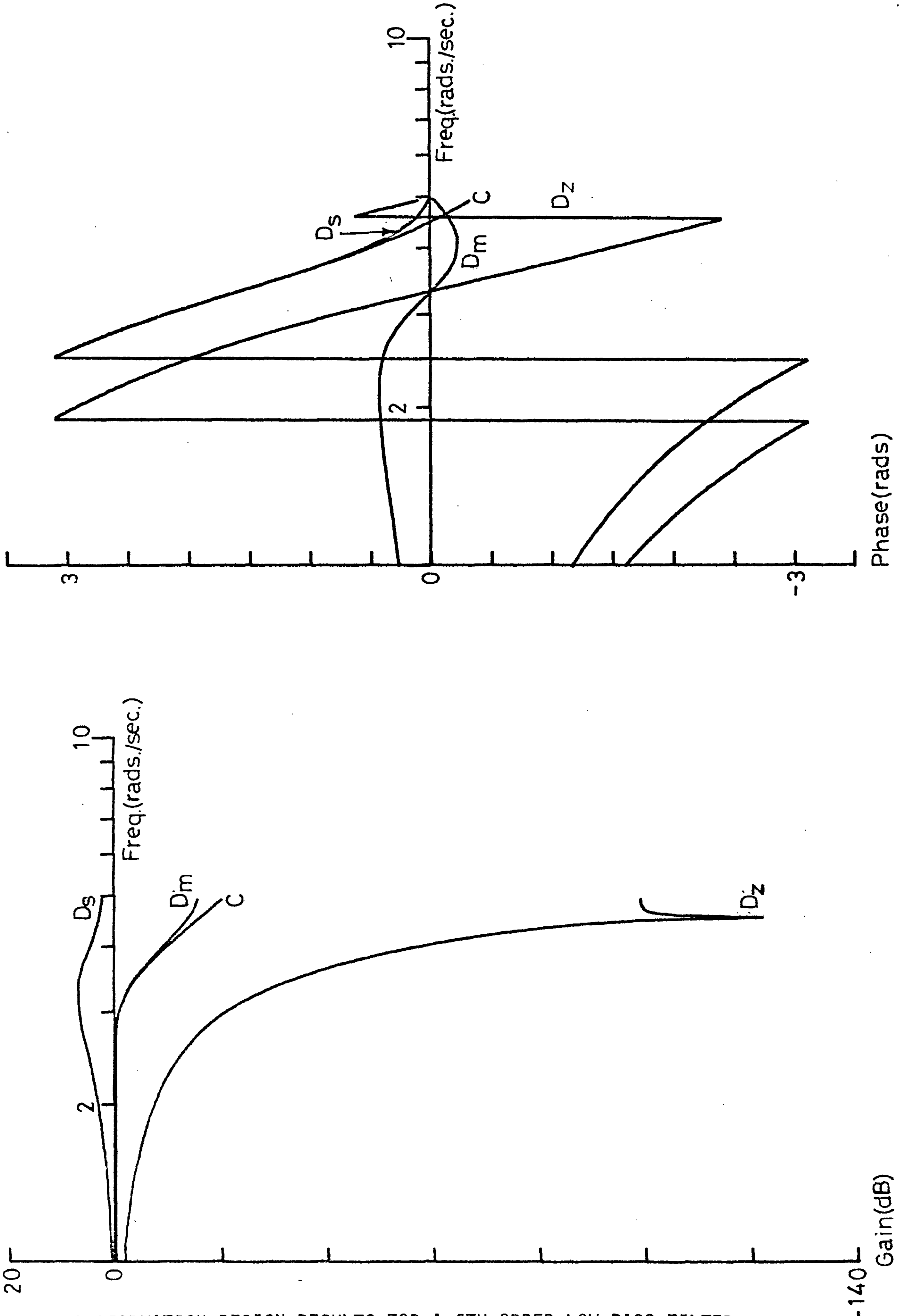
THE RELATIONSHIPS OF INSERTED ZEROS TABLE 4.1



ZERO INSERTION GAIN RESPONSES FIGURE 4.1

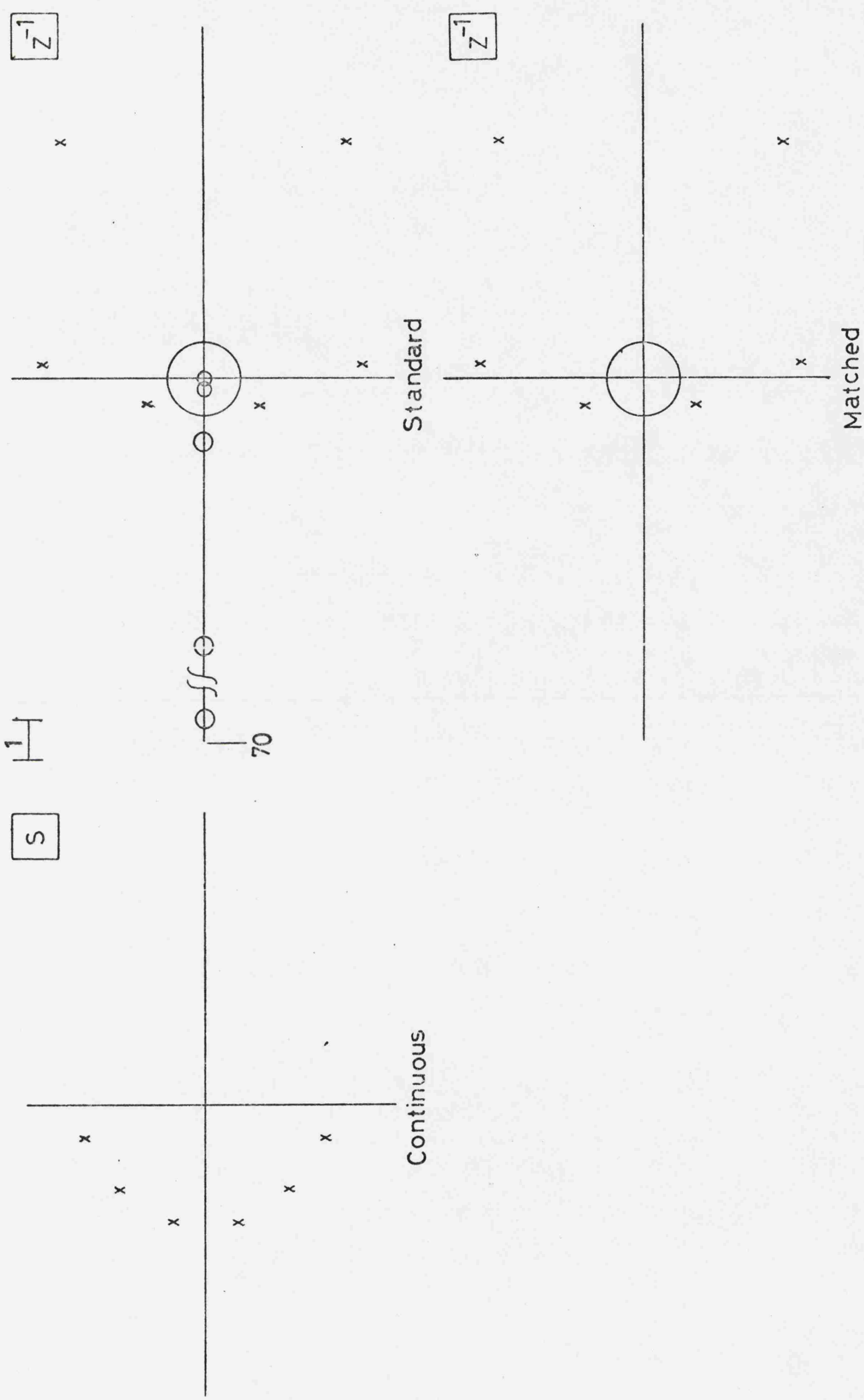


THE FOURTH ORDER CHEBYSHEV RESULT FIGURE 4.2

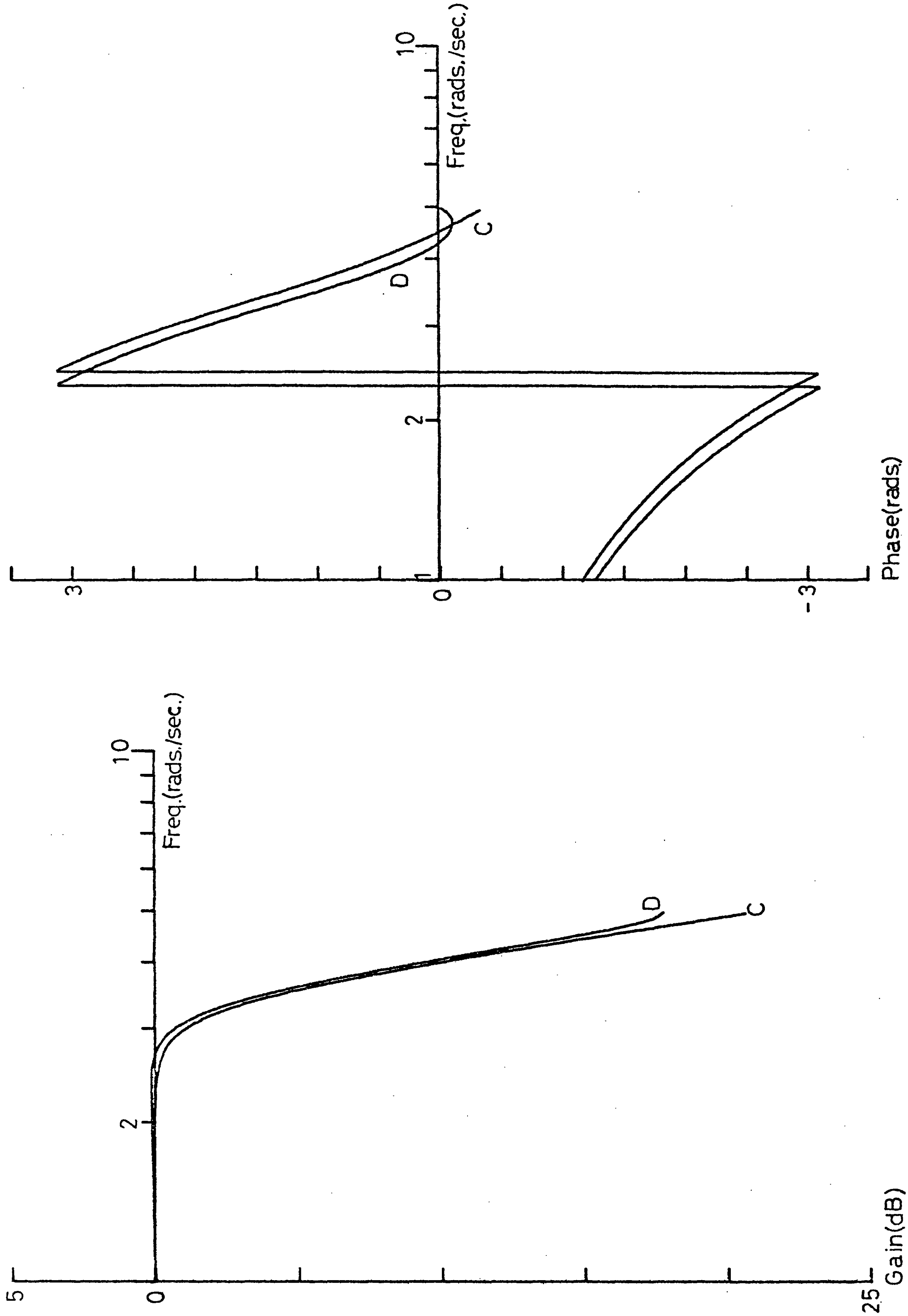


THE TRANSFORMATION DESIGN RESULTS FOR A 6TH ORDER LOW-PASS FILTER

FIGURE 4.3

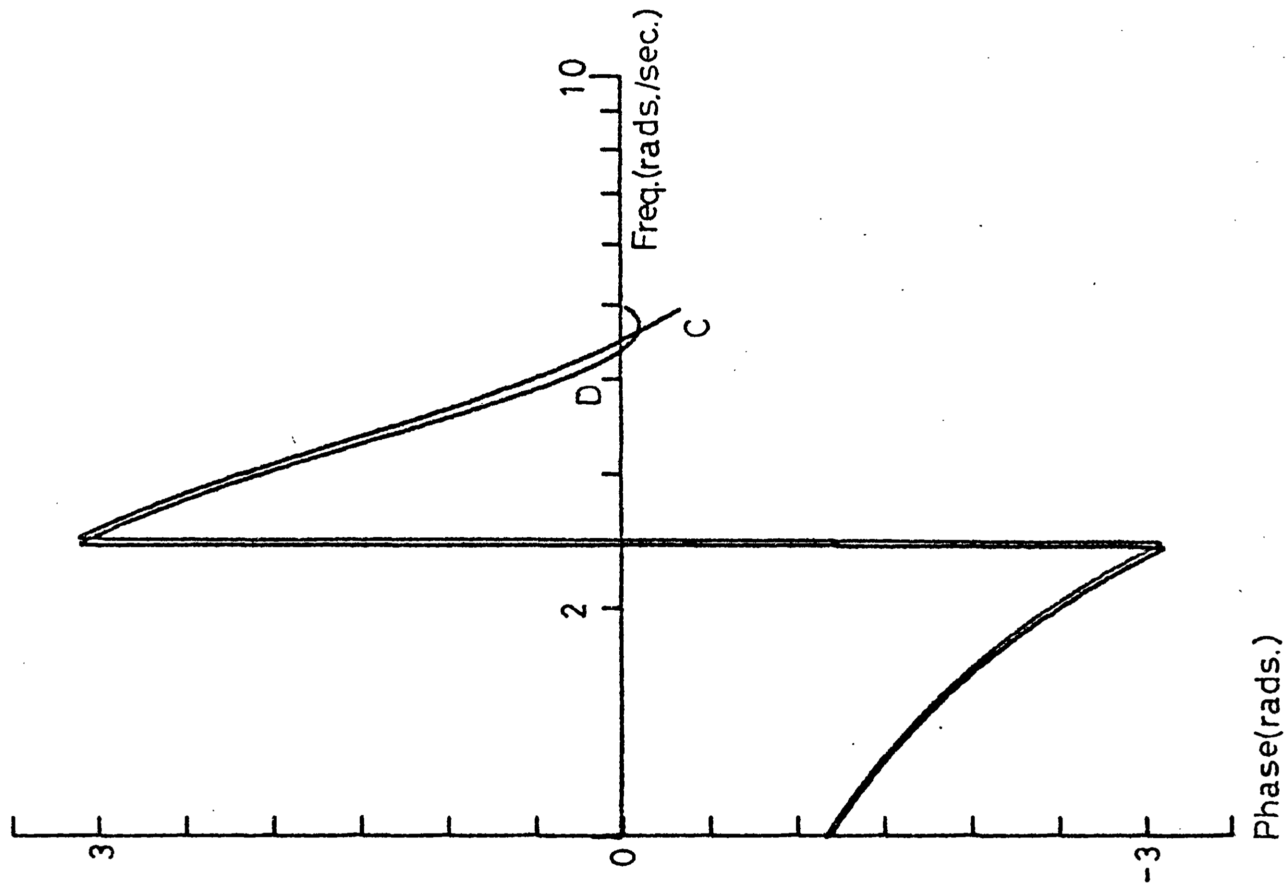


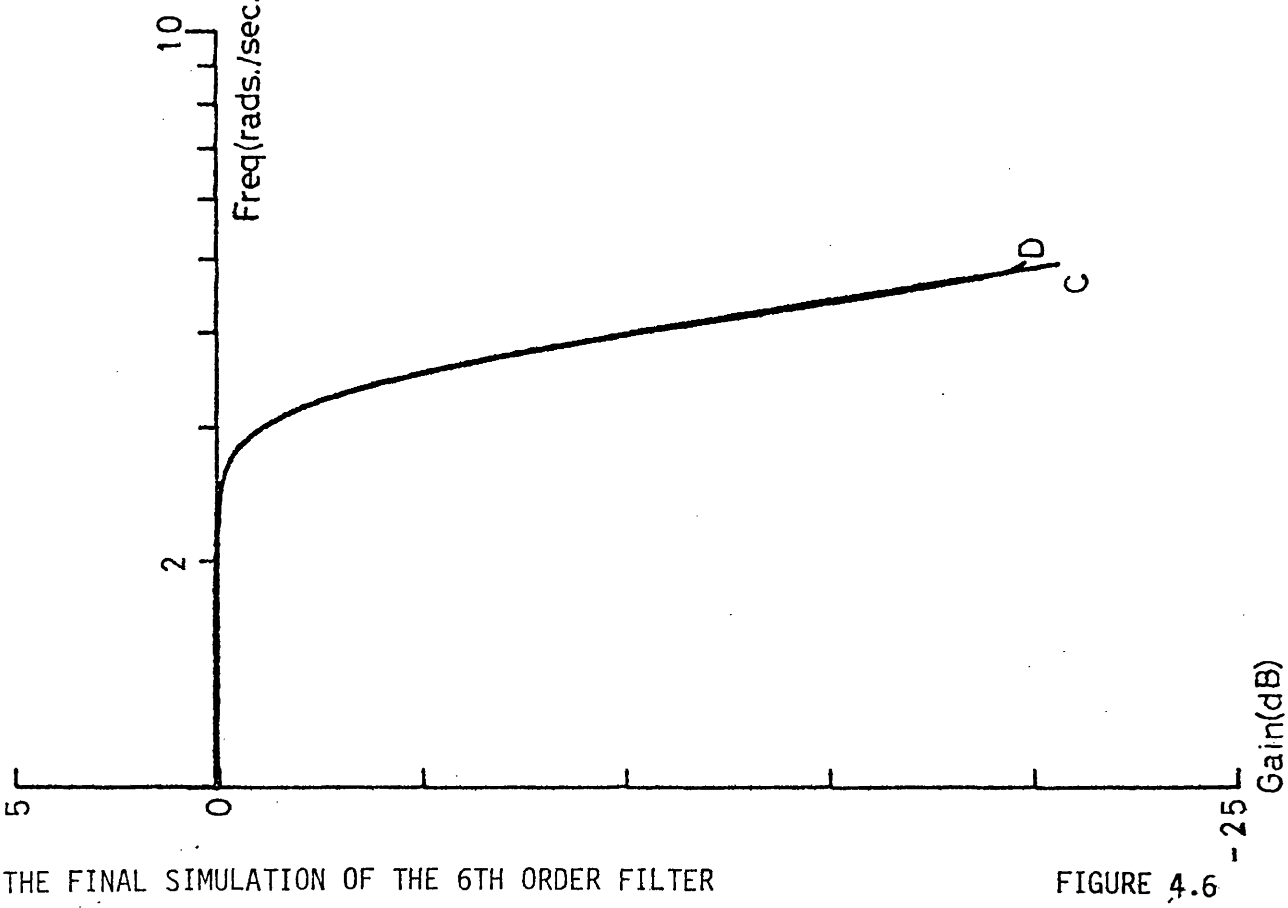
POLE-ZERO PLOTS FOR THE 6TH ORDER LOW-PASS FILTER FIGURE 4.4



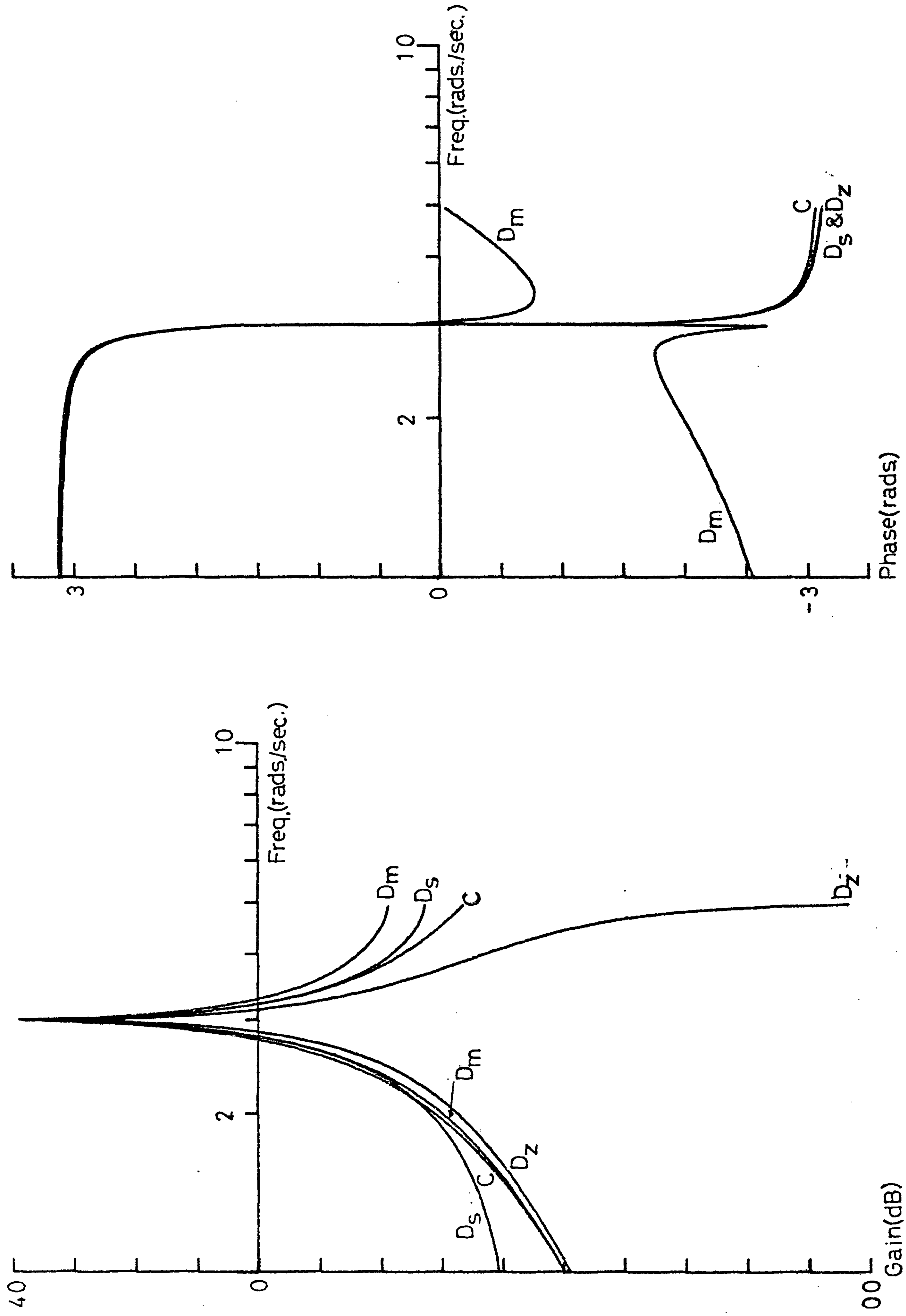
THE 'ROOT-SEARCHING' SIMULATION OF THE 6TH ORDER LOW-PASS FILTER

FIGURE 4.5



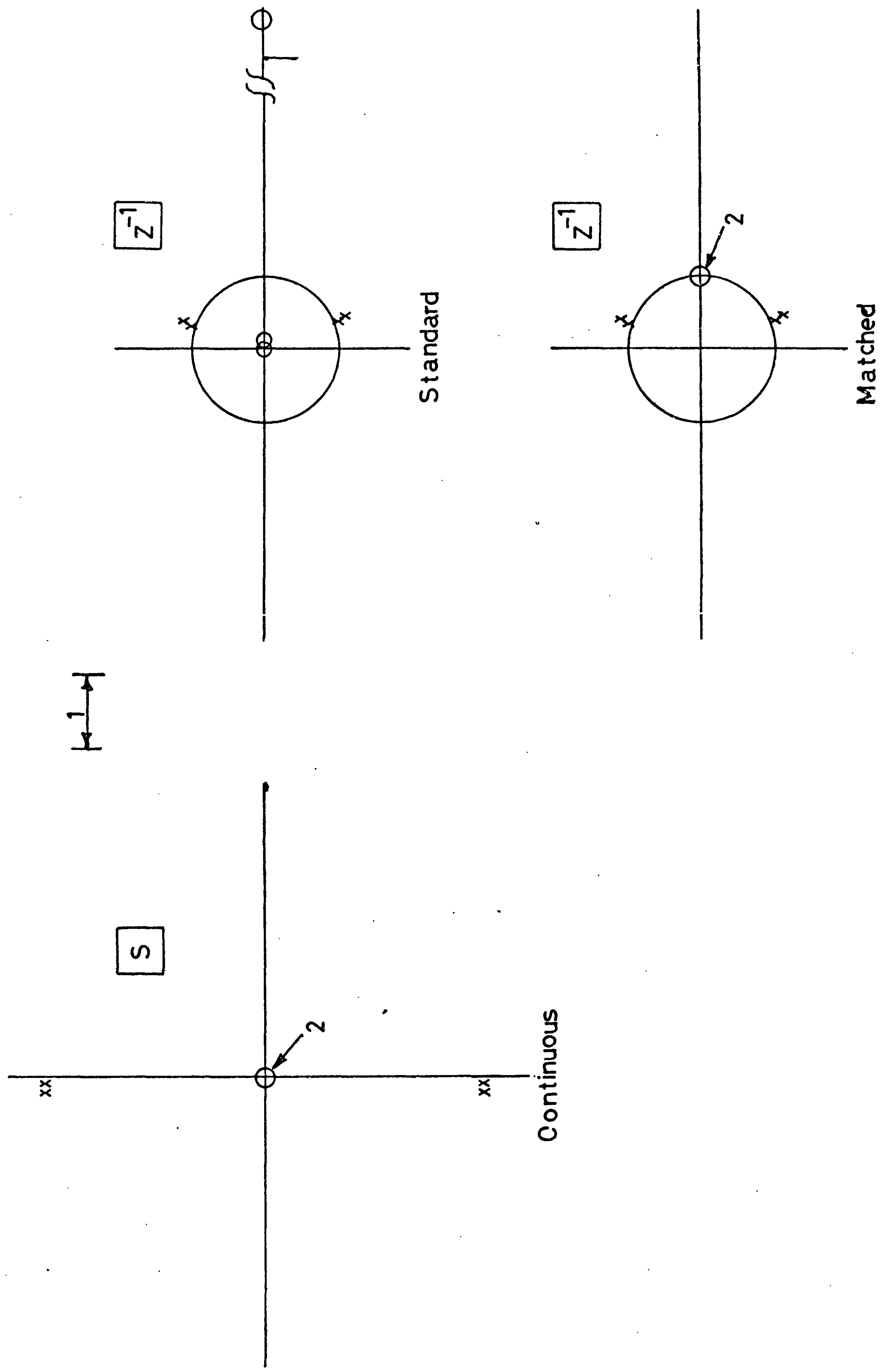


THE FINAL SIMULATION OF THE 6TH ORDER FILTER FIGURE 4.6



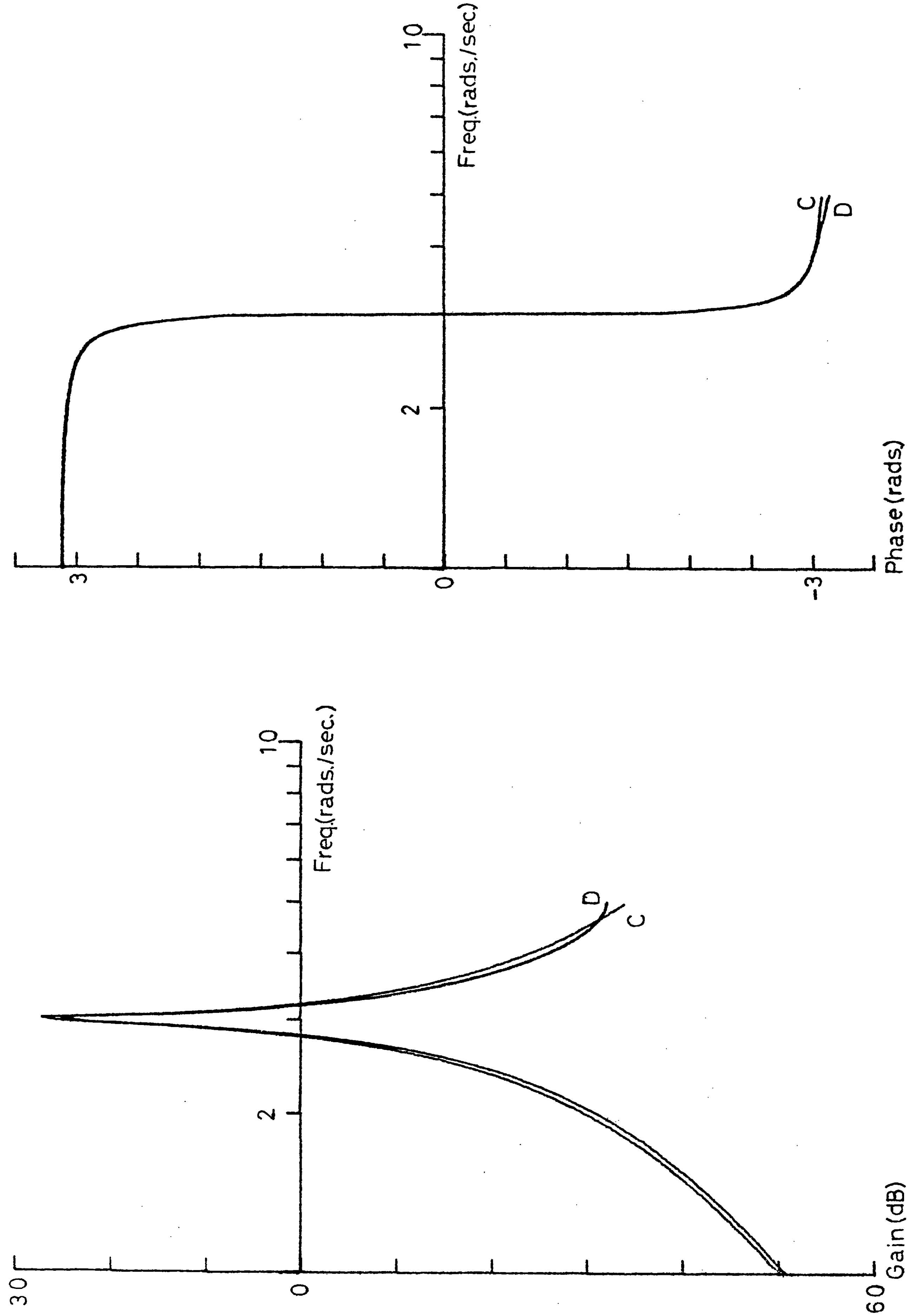
THE TRANSFORMATION DESIGN RESULTS FOR A BAND-PASS FILTER

FIGURE 4.7



POLE-ZERO PLOTS FOR THE BAND-PASS FILTER

FIGURE 4.8



A SIMULATION DESIGN RESULT FOR THE BAND-PASS FILTER

FIGURE 4.9

5. CONSIDERATIONS OF THE DIGITAL UNIT PULSE RESPONSE

The zero insert method showed that simulation filters can be designed to have reasonable frequency response accuracy. This result encourages the study of other methods of approach to digital filter design. Other approaches are also necessary in view of the specification considerations discussed in Section 1.1, only 's' plane specifications having been used so far. The object of this section of the work is to study the use of the unit pulse response as a basis for filter design.

The Standard Z Transform is based upon a defined equivalence of the sampled continuous impulse response and the digital unit pulse response. Chapter 2 showed the limitations of the Standard Z Transform in terms of frequency response performance, by way of the serious alias error that can result. However, the studies also showed that acceptable simulation designs can result when using the Standard Z Transform. It is the intention here to study the use of the unit pulse response in a more general sense, and to relate the unit pulse response to the desired frequency response thus hoping to produce good simulations.

5.1 Filter Design from the Unit Pulse Response

Let the transfer function, in Z^{-1}, of a digital filter (whose unit pulse sequence has members h_n) be:

$$H(Z^{-1}) = \frac{\sum_{i=o}^{I} a_i Z^{-i}}{1 - \sum_{k=1}^{K} b_k Z^{-k}} \qquad 5.1$$

Then:

$$h_n = a_n + \sum_{k=1}^{K} b_k\, h_{n-k}, \qquad 5.2$$

also assume that the filter is causal, that is:

$$h_n = 0 \text{ for, } n < 0\ . \qquad 5.3$$

Thus the elements of the unit pulse response become:

$$\begin{aligned} h_o &= a_o, \\ h_1 &= a_1 + b_1\, h_o, \\ h_2 &= a_2 + b_1\, h_1 + b_2\, h_2, \\ &\vdots \\ h_I &= a_I + b_1\, h_{I-1} + \text{-----------}\ b_p\, h_{I-p}, \\ &\vdots \\ h_q &= b_1\, h_{q-1} + b_2\, h_{q-2} + \text{------------}\ b_K\, h_{q-K}, \end{aligned} \qquad 5.4$$

where $p = I$ for $I < K$,

$p = K$ for $I \geq K$

and where $q \geq K$.

Any set of such equations where $q \geq K$ forms a set of linear equations in the variable b_k. If ℓ equations are formed where $\ell \geq K$, and if the unit pulse sequence h_n is known then the equations can be solved for the values of b_k. Expressed in matrix form the equations are:

$$\begin{pmatrix} h_{q-\ell} \\ \vdots \\ h_q \end{pmatrix} = \begin{pmatrix} h_{q-\ell-1} & h_{q-\ell-2} & \cdots & h_{q-\ell-K} \\ \vdots & & & \vdots \\ h_{q-1} & \cdots & \cdots & h_{q-K} \end{pmatrix} \begin{pmatrix} b_1 \\ \vdots \\ b_K \end{pmatrix} \qquad 5.5$$

Therefore, given a causal, rational unit pulse sequence h_n it is possible to find the values b_k and hence by back substitution the values a_i thereby synthesising $H(Z^{-1})$.

5.2 Using Frequency Response Data

The aim of this work is to produce an accurate frequency response model. The unit pulse response of a digital filter can be found from its frequency response by the use of the inverse Z transform (IZT). The inverse Z transform is defined:

$$y(nT) = \frac{1}{j2\pi} \oint Y(Z)Z^{n-1}dZ, \qquad 5.6$$

where the path of integration is within the region of convergence of the infinite sequence $y(nT)$. For this application the path of integration would be the unit circle in the Z plane. Now consider the discrete Fourier transform (D.F.T.) defined:

$$X_n = \frac{1}{N} \sum_{k=o}^{N-1} x_k \, e^{-j\frac{2\pi nk}{N}}, \qquad 5.7$$

and the inverse (IDFT):

$$x_k = \sum_{n=o}^{N-1} X_n \, e^{j\frac{2\pi nk}{N}}. \qquad 5.8$$

The summation of Equation 5.8 approximates to the integration of Equation 5.6, both of which operate on the unit circle. As N tends to infinity Equation 5.8 tends to Equation 5.6. Therefore the well known result is seen, namely that the IDFT is a sampled approximation to the IZT. The degree of approximation involved in using the IDFT is dependent upon the number of unit circle samples N. The importance of this observation is that the IDFT is easily calculable by use of a digital computer, whereas the IZT is not. Now, if Y(Z) defines a digital filter then y(nT) is the unit pulse response of that filter. Thus if X_n is the sampled version of the frequency response Y(Z) then x_k approximates $y(nT)$. In practical terms the validity of the approximation is dependent upon the length of the sequence $y(nT)$. If $y(nT)$ is of significant amplitude for $n > N$ then the truncation effect of Equations 5.7 and 5.8 invalidates the approximation.

The method as described so far essentially requires the frequency response to be rational in terms of the Z^{-1} plane variable. It has been seen (from Section 3) that continuous frequency responses are often not realisable in terms of a rational transfer function in Z^{-1}. The simple use of a truncated continuous frequency response prior to application of the IDFT can produce impractical unit pulse responses. The responses can be impractical in the sense that they may be non-causal and also require very high order simulations. For economic models the requirement is that the model should be of similar order to the original element and should be causal. The above limitations mean that, in practical circumstances, some approximation to the required frequency response will need to be considered to produce a response that is more 'sensible' in terms of the digital domain.
(Note: more 'sensible' here means, approaching rationality without necessarily producing a rational frequency response).
Methods of approximation will be considered in later sections.

To conclude, if the desired frequency response of a digital filter is known, it is possible to calculate an equivalent transfer function $H(Z^{-1})$, subject to certain limitations. The process of this calculation is to take the IDFT of the frequency response and solve the equations formed by the resultant unit pulse response as described in Section 5.1. The next section will establish the feasibility of this process.

5.3 The Feasibility of Using Unit Pulse Response Methods

It has been stated that it is necessary to modify the desired frequency response such that it is more rational with respect to Z^{-1}. An obvious adaptation that could be considered is to modify the frequency response such that it becomes the response of a known transformation method. This proposal is of no great value for simulation, but is a worthwhile consideration as a test for the algorithm developed so far. Consider using the frequency response of the equivalent Bilinear Z Transform designed digital filter. Then let X_n be calculated from the original function such that:

$$X_n = H(s)\Big|_{s = j\frac{2}{T}\tan\frac{\pi n}{N}} \qquad 5.9$$

using Equation 2.19 to determine the position of the bilinearly warped frequency samples. Consider a simple example, let:

$$H(s) = \frac{1}{s^2 + \sqrt{2}\, s + 1} \qquad 5.10$$

a Butterworth filter. In addition let the sampling frequency be 10 radians/second and the IDFT be taken at 128 points. The values of the unit pulse response thus produced are found in Table 5.1. Then, setting up linear equations as in Equation 5.5, and solving for the filter coefficients gives:

$$G(Z^{-1}) = \frac{0.06396(1 + 2Z^{-1} + Z^{-2})}{1 - 1.16832^{-1} + 0.42413\ Z^{-2}} . \qquad 5.11$$

From Equation 2.21 the Inverse Bilinear Z Transform is:

$$Z^{-1} \Rightarrow \frac{1 - sT/2}{1 + sT/2} . \qquad 5.12$$

Now, $T = 2\pi/10$. Therefore using Equation 5.12 to transform Equation 5.11 gives:

$$H'(s) = \frac{1}{s^2 + 1.4142\ s + 1} . \qquad 5.13$$

Thus:

$$H'(s) \approx H(s), \qquad 5.14$$

Q.E.D. Thus it has been shown that for a rational frequency response, a digital filter can be designed by the use of the IDFT and the unit pulse design method.

It is an interesting side issue to note that the procedure used in the above example amounts to a practical method of deriving a continuous system transfer function from its frequency response. The procedure for such a method is thus:

(i) obtain N values of the frequency response such that they represent the sampled frequency response of a bilinearly designed digital filter;

(ii) apply this data to the IDFT;

(iii) solve the resulting linear equations formed from the unit pulse response;

(iv) use the Inverse Bilinear Z Transform to provide the continuous transfer function.

The application of this method is almost entirely restricted to implementation by use of a digital computer. Investigation of the literature has shown that this procedure is very similar to, and can be regarded as a practical algorithm for, the Wiener- Lee transforms (Ref. 44). The Wiener-Lee transforms use a trigonometric tangent warping technique to produce a periodic frequency response. The Fourier series of the periodic frequency response is used to find the resultant transfer function.

5.4 Unit Pulse Response Methods for Simulation Filters

The method developed above was interesting and valuable in that it provided a test of the theoretical basis, but is not of direct application to the problem in hand. To return therefore to the real problem. The truncated continuous frequency response needs modification. The modification must be such that the resultant response closely approximates the original response, yet becomes more 'sensible' in terms of a Z^{-1} plane frequency function. References 25, 38, 44 and 59 are typical of much literature concerned with the subject of frequency response relationships and approximation methods. Various methods are investigated in the literature, and it is now intended to investigate one of these methods which seems particularly applicable to the problem in hand. This method consists of rationally relating the magnitude and phase responses by use of the Hilbert transform.
To summarise the literature, defining the response:

$$H(\omega) = e^{-\alpha(\omega) - j\theta(\omega)} , \qquad 5.15$$

then

$$\theta(\omega_0) = \frac{\omega_0}{\pi} \int_{-\infty}^{\infty} \frac{\alpha(\omega)}{\omega^2-\omega_0^{\ 2}} d\omega \ , \qquad 5.16$$

and

$$\alpha(\omega_0) = \alpha(o) - \frac{\omega_0^{\ 2}}{\pi} \int_{-\infty}^{\infty} \frac{\theta(\omega)}{\omega(\omega^2-\omega_0^{\ 2})} \ , \qquad 5.17$$

defining the magnitude response,

$$A(\omega) = e^{-\alpha(\omega)} \ , \qquad 5.18$$

then,

$$\theta(\omega_0) = \frac{\omega_0}{\pi} \int_{-\infty}^{\infty} \frac{\ln A(\omega)}{\omega^2-\omega_0^{\ 2}} d\omega \ . \qquad 5.19$$

Equation 5.19 being the Hilbert transform relationship. The Hilbert transform is easily applied by use of a Fourier Transform. Consider the Fourier transform of a function:

$$A(y) = FT\{a(x)\} \qquad 5.20$$

then the Hilbert transformed version of a(x) is found from the inverse Fourier transform of:

$$B(y) = -j \text{ sign } (y).\ A(y). \qquad 5.21$$

The literature shows that the above relationships define realisable transfer functions. It should be noted that $\theta(\omega)$ is minimum phase for the necessary condition that H(s) is analytic in the right hand half s plane. If now the application of the Hilbert transform is considered in the Z^{-1} plane, then this can be accomplished by using the DFT instead of the Fourier transform of Equations 5.20 and 5.21. In accordance with Equation 5.19 the Hilbert transform is applied to $\ln A(\omega)$ and a

minimum phase response results. The modified phase response, having been found, is 'sensibly' related to the magnitude response with respect to the Z^{-1} plane frequency variable. Essentially this procedure forces a phase response which is compatible with the defined magnitude response in terms of the Z^{-1} plane. Practically of course the new response is still not ideal due to the necessary discontinuity in the slope of the magnitude response (see Chapter 3), however this has not been found to be disruptive in the production of reasonable simulation filters.
As stated, the resultant frequency response is minimum phase and so all of the zeros will be outside the unit circle in the Z^{-1} plane.
Section 4.1 investigates Z^{-1} plane zeros and shows that non-minimum phase zeros can be created by simply inverting the zero position.
That is, for a minimum phase zero placed at:

$$Z^{-1} = -a \text{ (say)},$$

then the equivalent non-minimum phase zero is placed at:

$$Z^{-1} = -1/a.$$

If the minimum phase response is $-\phi(\omega)$ then the non-minimum phase response is:

$$\theta(\omega) = \phi(\omega) - \omega T. \qquad 5.22$$

For complex conjugate zeros, both zeros must be inverted and if the phase of the zero pair is $-\phi(\omega)$ then the non-minimum phase response is:

$$\theta(\omega) = \phi(\omega) - 2\omega T. \qquad 5.23$$

For an example with large numbers of zeros it will be necessary to modify groups of zeros such that the closest approximation to the desired phase response can be found. The proposed method is limited in that there is difficulty in representing certain types of roots:

(a) Where roots lie on the frequency axis (invariably zeros) the problems arise due to the samples of the frequency response not coinciding with the roots. Problems also arise where such roots are accurately represented in that it is difficult to take the logarithm of the resultant magnitude.

(b) Where complementary pole-zero pairs occur in the original response (phase equalisation) the minimum phase criterion eliminates these roots from the resultant filter.

These problems together with the problems discussed earlier must be dealt with by the overall algorithm. Additionally some method must be introduced to check the final result against the original response. In practice accurate matching of the magnitude response is all that can be expected from this method, as the phase is determined from the magnitude response by use of the Hilbert transform. In practice if the modelling of the magnitude response was not good enough, then it was found to be useful to increase the order of simulation. Such increases in order effectively add some equalisation to the final response. Further equalisation of the phase response may be necessary in certain cases to provide a more accurate overall result. Any increases in the order of the model must be carefully considered as they may be costly in terms of the time required to execute a simulation.

5.5 An Overall Algorithm

The flow diagram of the whole algorithm is shown in Figure 5.1. The algorithm implements the procedure as discussed and has features which eliminate the problems mentioned. The steps of the flow diagram are numbered to facilitate discussion of the algorithm. The reasons for certain features are not entirely obvious, these items are discussed below.

(a) Steps 1 to 6 find a digital filter which is equivalent to a filter designed by the Bilinear Z Transform, in a similar way to the example in Section 5.3. The object of this exercise is to find a digital filter from a definitely rational function of Z^{-1}. This digital filter transfer function can then be used to identify pole and zero positions which may be eliminated or not accurately represented by the later analysis. Phase equalising and unit circle roots are those likely to suffer. If the original response is specified by a transfer function in s, then steps 1 to 7 can be replaced by the Matched Z Transform.

(b) Steps 10 to 16 form the minimum phase unit pulse response which is to be used to design the simulation filter.

(c) Steps 17 to 20 design and test the filter and insert the relevant roots from the earlier 'Bilinear' analysis.

(d) Step 3 simply states that the unit pulse response should be valid. This test is designed to ensure that the value of N is sufficiently large to ensure that the unit pulse response is not truncated when of significant amplitude.

(e) Steps 4 and 17 form simultaneous equations from unit pulse response data. This process requires knowledge of the proposed filter orders. In the algorithm as described the order is entered by the user. In a practical algorithm in a final simulation system a first approximation to the order would have to be derived from the frequency response data.

(f) Step 20 states that the result must be checked. The question of how accuracy is determined is difficult to make explicit. Certain users may well wish to define limits of accuracy in different ways. Least square or limits of deviation are examples of methods that could be used to determine the accuracy of the simulation. The testing method used for the examples (see later) in fact checked the accuracy by checking the deviation of the root positions from those found from the 'Bilinear' section of the algorithm. This method is useful for demonstration purposes, but is not to be recommended for practical situations. The problem with this method is that certain filter designs can fail the test and yet be reasonable simulations. The example shown in Figures 5.3 and 5.4 is a classic example of this problem, in that the filters designed failed the root deviation test and yet are reasonable simulations. For practical simulation design situations a method should be implemented which checks the actual frequency response according to an error criterion.

5.6 Examples

The examples for the unit pulse response design method are shown in Figures 5.2 to 5.7. The figures show graphs of magnitude and phase versus frequency. The curves shown on the graphs give the continuous response (marked C) and the digital response or responses (marked D). In all examples the sampling frequency is 10 rads/sec.

(i) Figure 5.2 shows the responses for a low-pass filter. The continuous prototype is a fourth order Chebyshev of 3dB ripple and 1 rad./sec. cut-off frequency. The results of designs by the transformation methods are shown in Figures 2.1 to 2.4. The zero insertion method result is shown in Figure 4.2.

(ii) Figures 5.3 and 5.4 show the frequency response results for a third order Butterworth filter. The filter is low-pass with a cut-off frequency of 4 rads./sec. Figure 5.3 shows the frequency response comparison for a fourth order simulation and Figure 5.4 a fifth order simulation. The transformation design responses for this filter are shown in Figures 2.5 to 2.8.

(iii) Figure 5.5 demonstrates the responses for a second order, band-pass filter. The transfer function of the continuous filter is:

$$H(s) = \frac{s}{s^2 + 0.7s + 1}$$

Three digital filter responses are shown:

D1 is the Standard Z Transform result;

D2 is a third order simulation;

D3 is a fourth order simulation.

(iv) Figure 5.6 shows the results for a second order, low-pass Butterworth filter of cut-off frequency of 1 rad./sec.

(v) Figure 5.7 shows the results for a second order band elimination filter. The transfer function of the continuous filter is:

$$H(s) = \frac{s^2 + 1}{s^2 + s + 1}$$

These examples show that accurate simulation, of the magnitude frequency response of a continuous filter, is obtainable using the unit pulse response design method. The continuous phase responses are not so accurately simulated. The phase error is a necessary consequence of the phase response being artificially related to the magnitude response by means of the Hilbert transform. A more detailed discussion of the results will be found in Chapter 7, where the simulation design methods are discussed and compared.

n	x_n
0	0.06396
1	0.20265
2	0.27358
3	0.23367
4	0.15696
5	0.08427
6	0.03188
7	0.00150
8	-0.01177
9	-0.01438
10	-0.01181

THE 'BILINEAR' UNIT PULSE RESPONSE TABLE 5.1

Fig.5.1 The 'Unit Pulse' Response Method

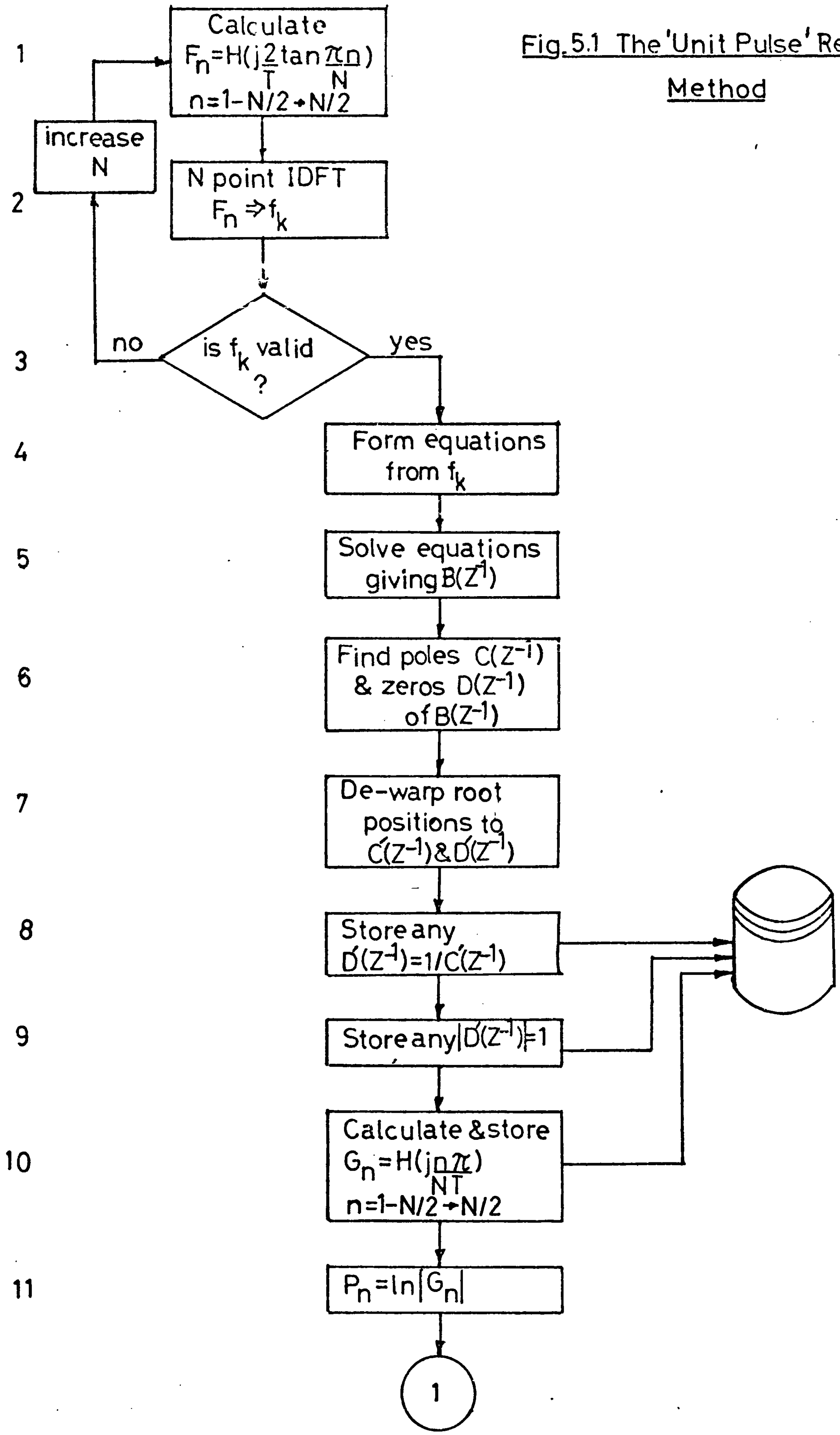
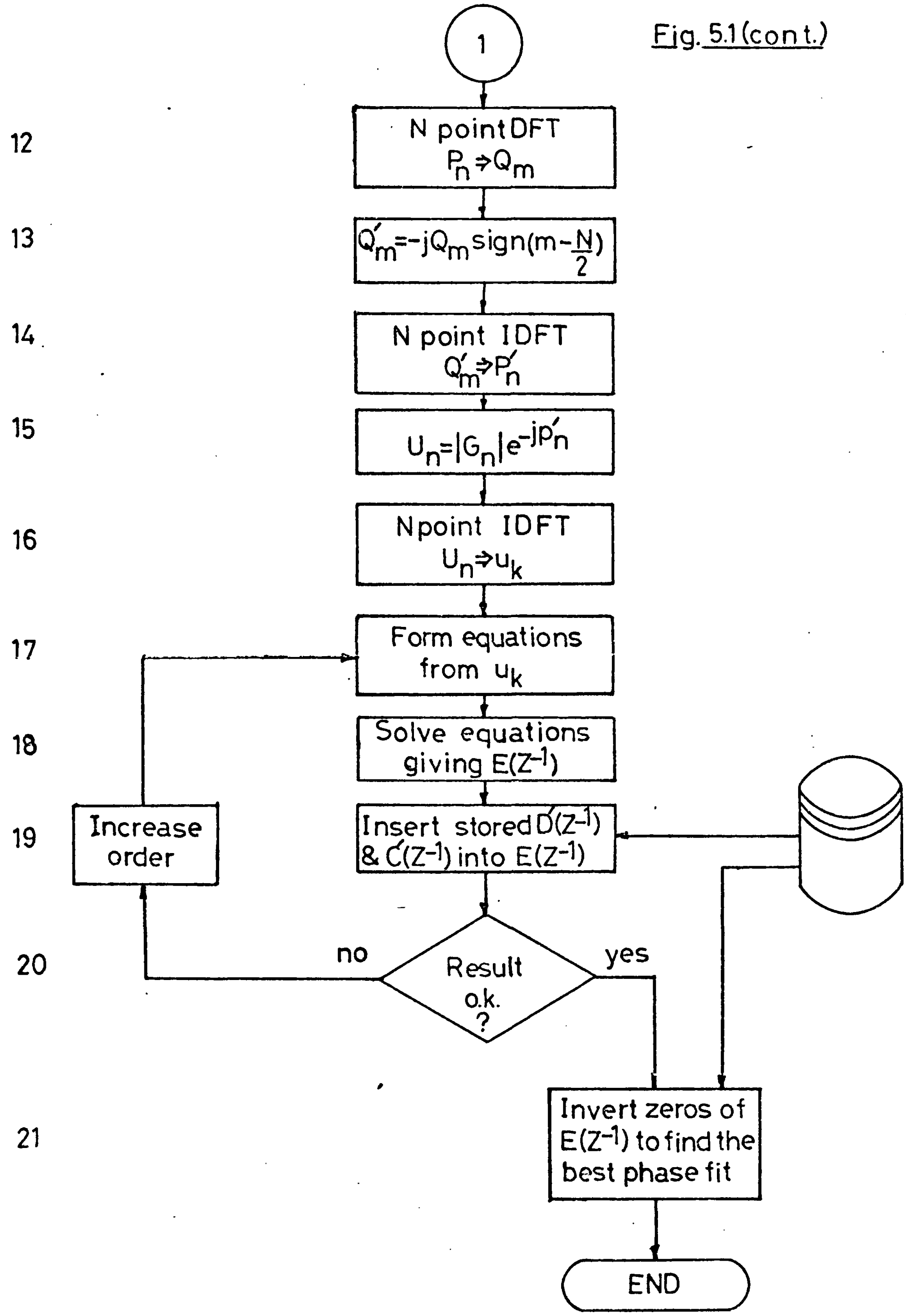
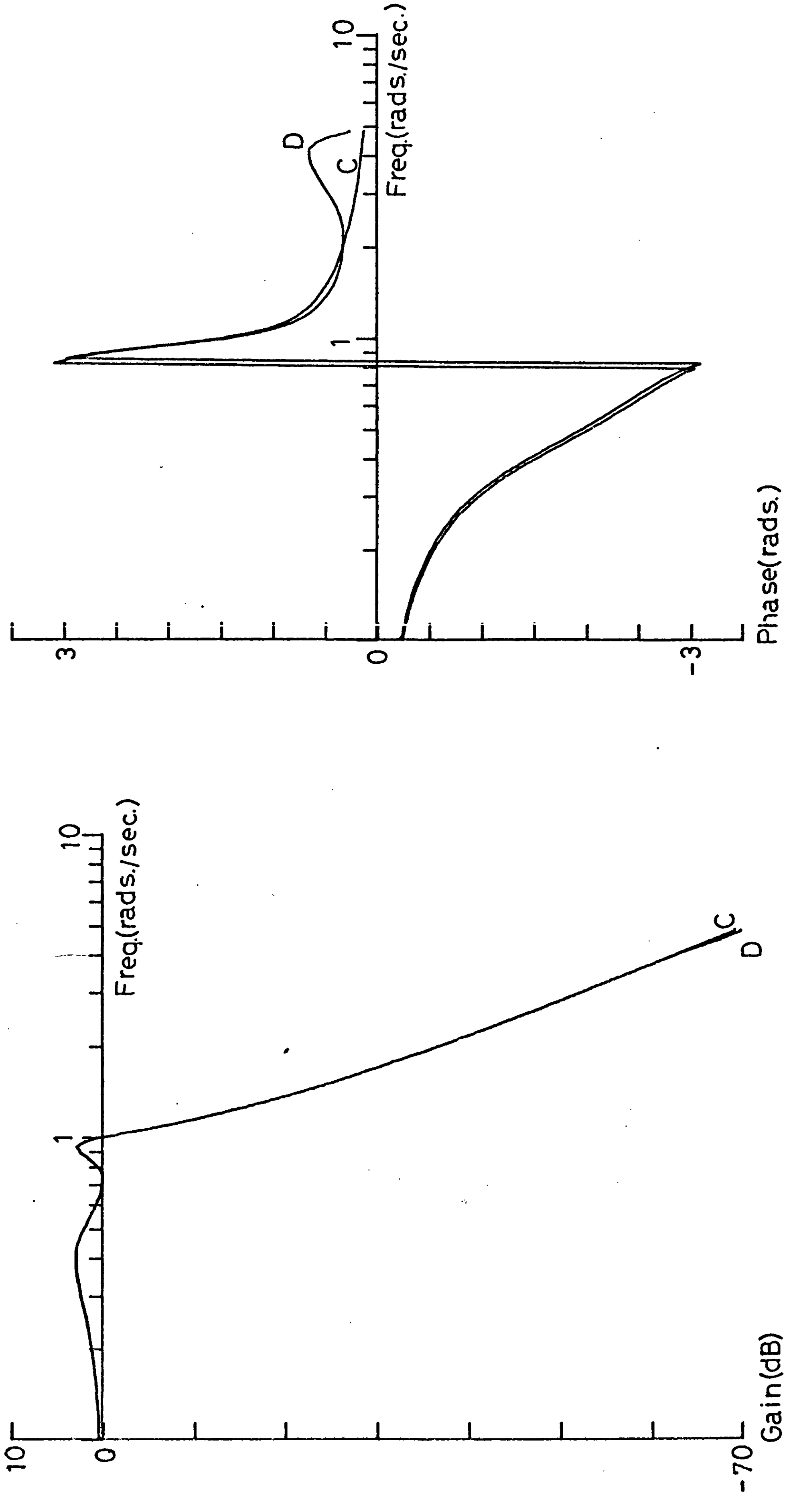
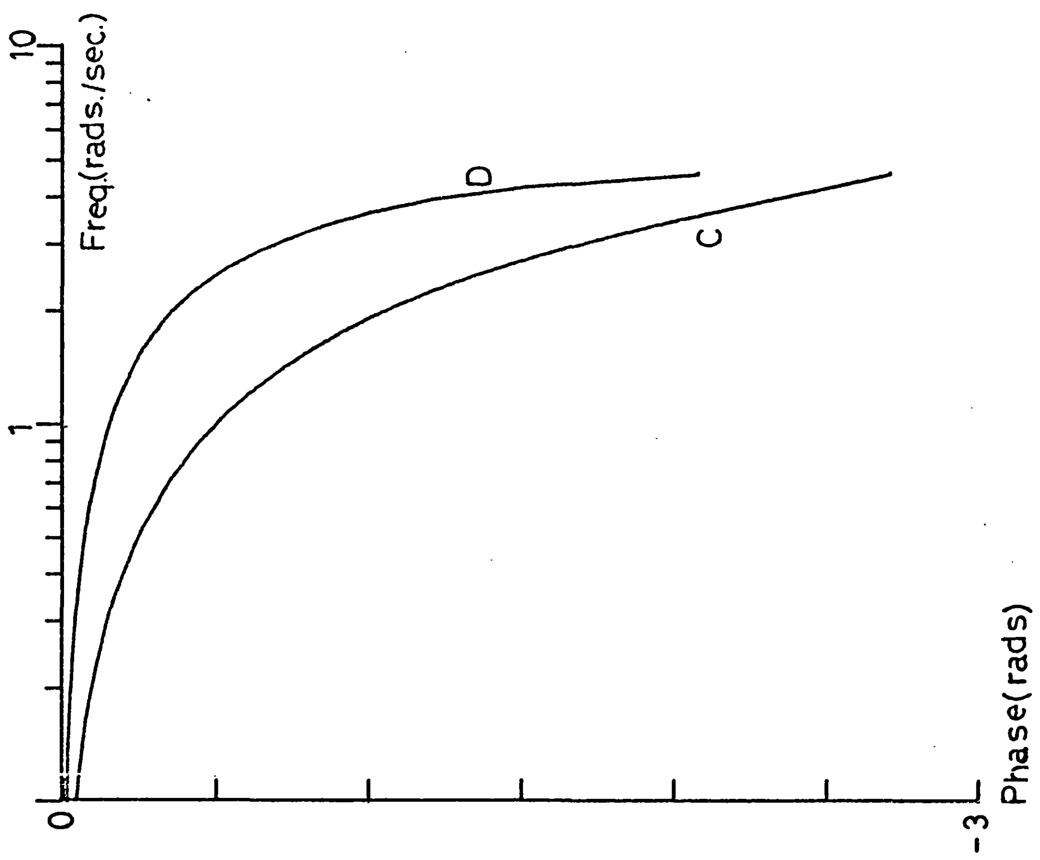


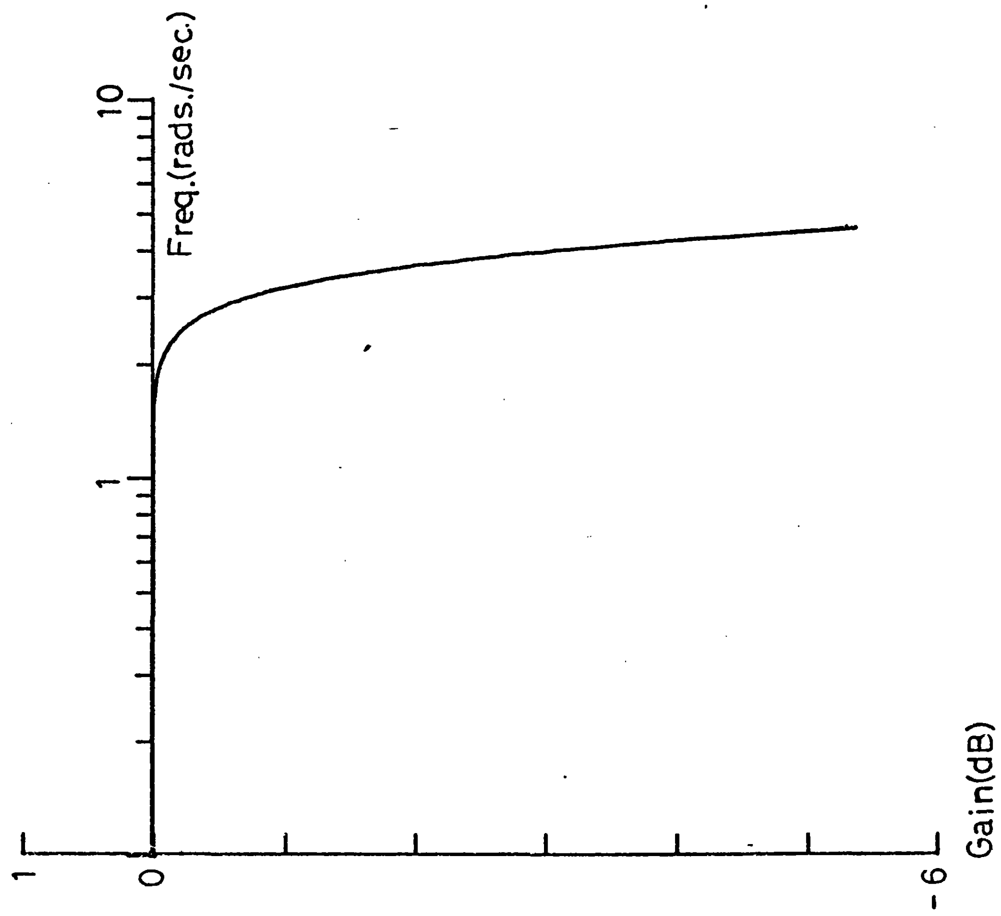
Fig. 5.1 (cont.)


1
12
N point DFT
$P_n \Rightarrow Q_m$
13
$Q'_m = -jQ_m \operatorname{sign}(m - \frac{N}{2})$
14
N point IDFT
$Q'_m \Rightarrow P'_n$
15
$U_n = |G_n| e^{-jp'_n}$
16
Npoint IDFT
$U_n \Rightarrow u_k$
17
Form equations from u_k
18
Solve equations giving $E(Z^{-1})$
19
Increase order
Insert stored $D'(Z^{-1})$ & $C'(Z^{-1})$ into $E(Z^{-1})$
20
no
Result o.k. ?
yes
21
Invert zeros of $E(Z^{-1})$ to find the best phase fit
END



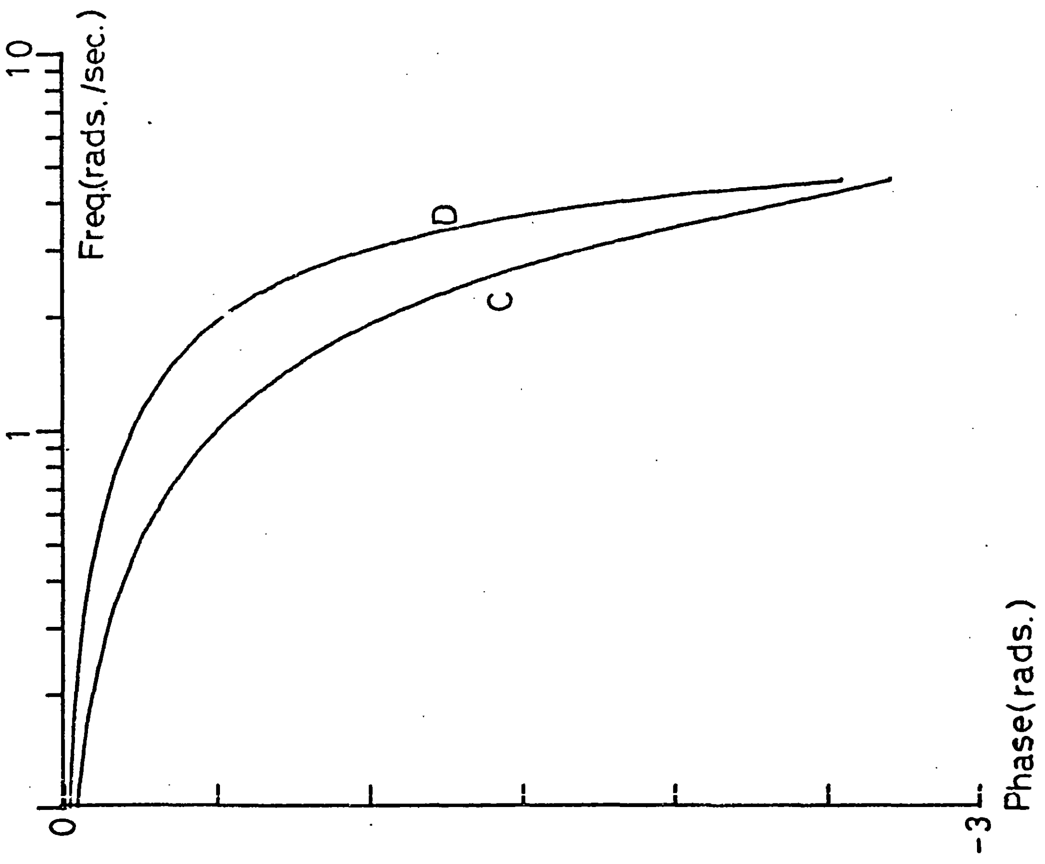
THE FOURTH ORDER CHEBYSHEV RESULT FIGURE 5.2

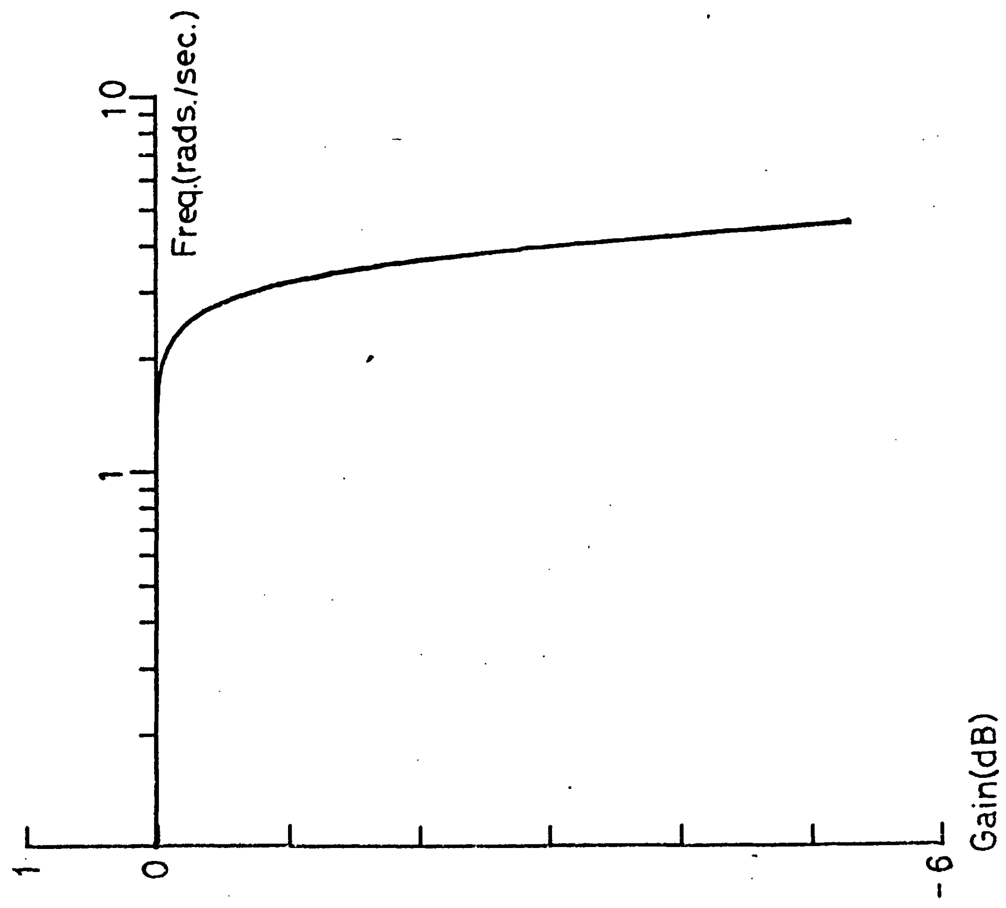




THE FOURTH ORDER SIMULATION OF THE 3RD ORDER BUTTERWORTH FILTER

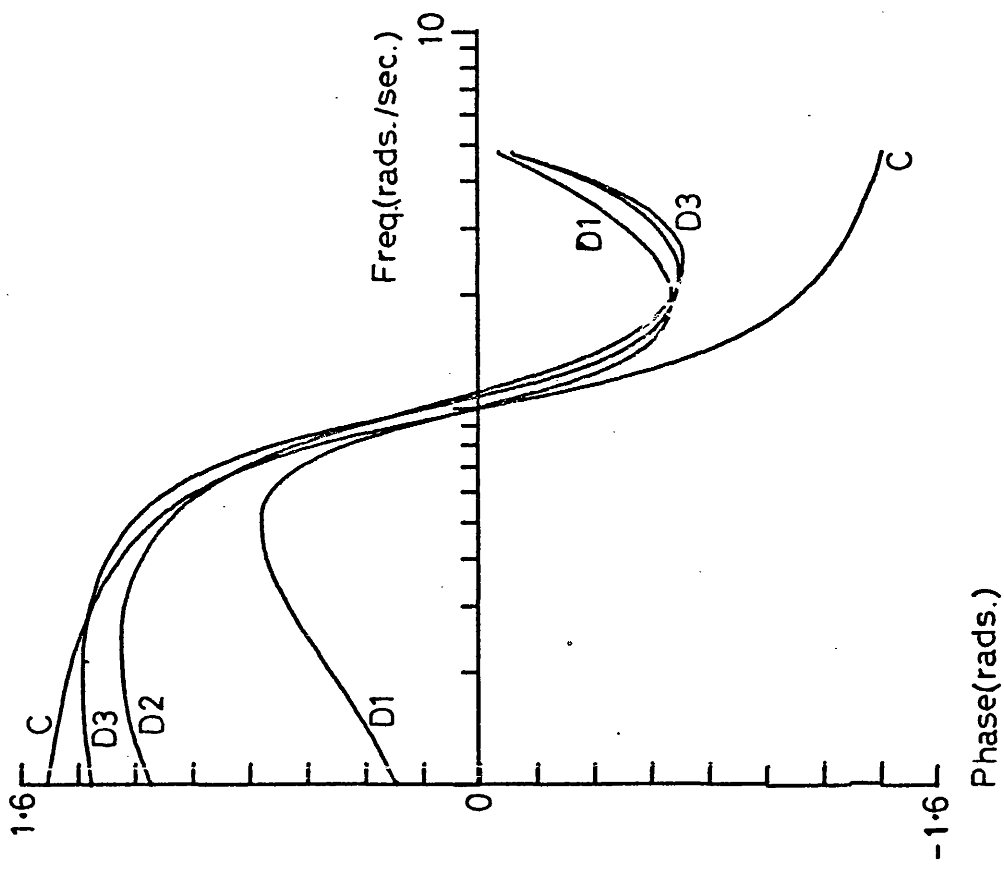
FIGURE 5.3

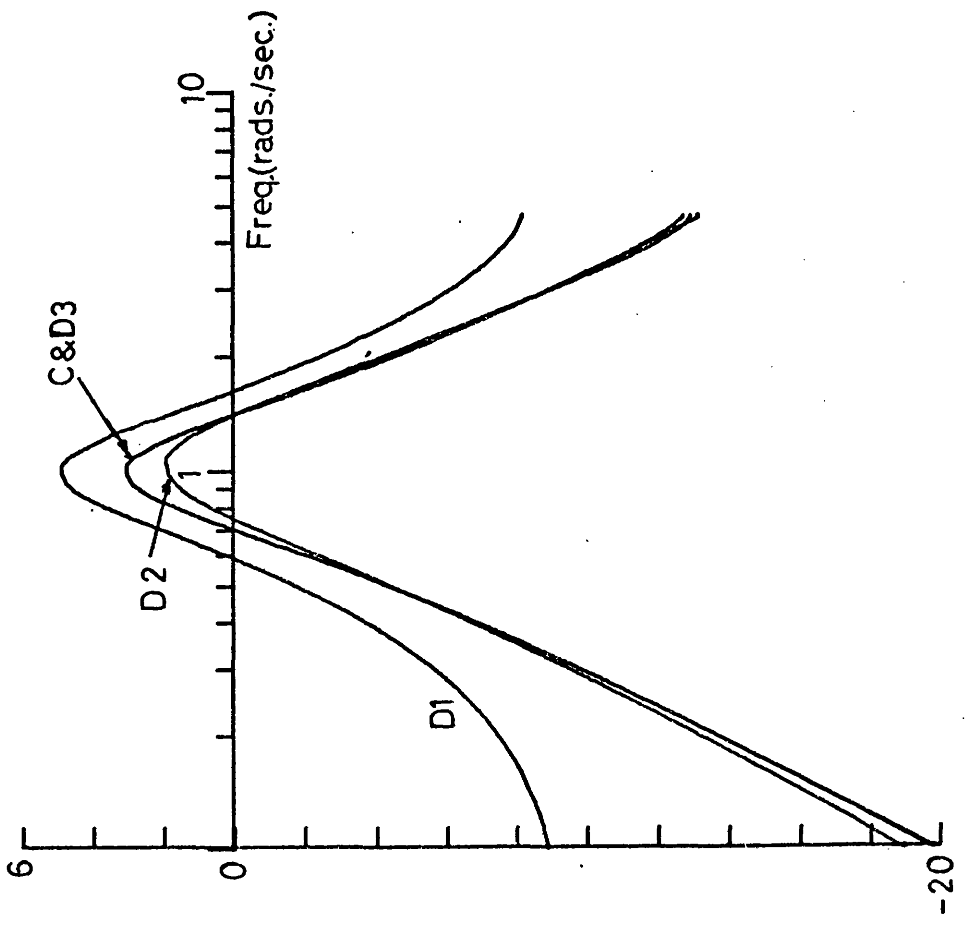




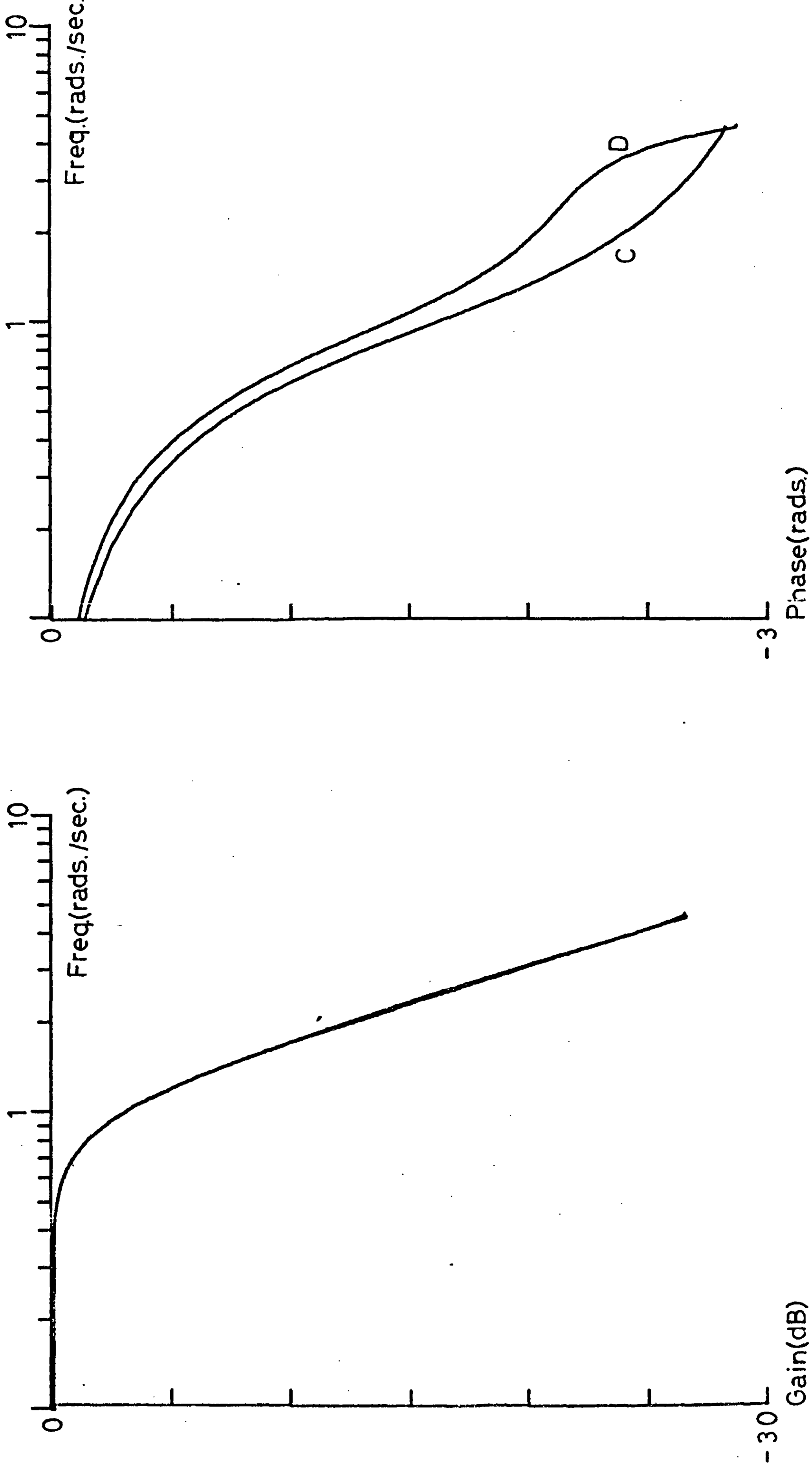
THE 5TH ORDER SIMULATION OF THE 3RD ORDER BUTTERWORTH FILTER

FIGURE 5.4



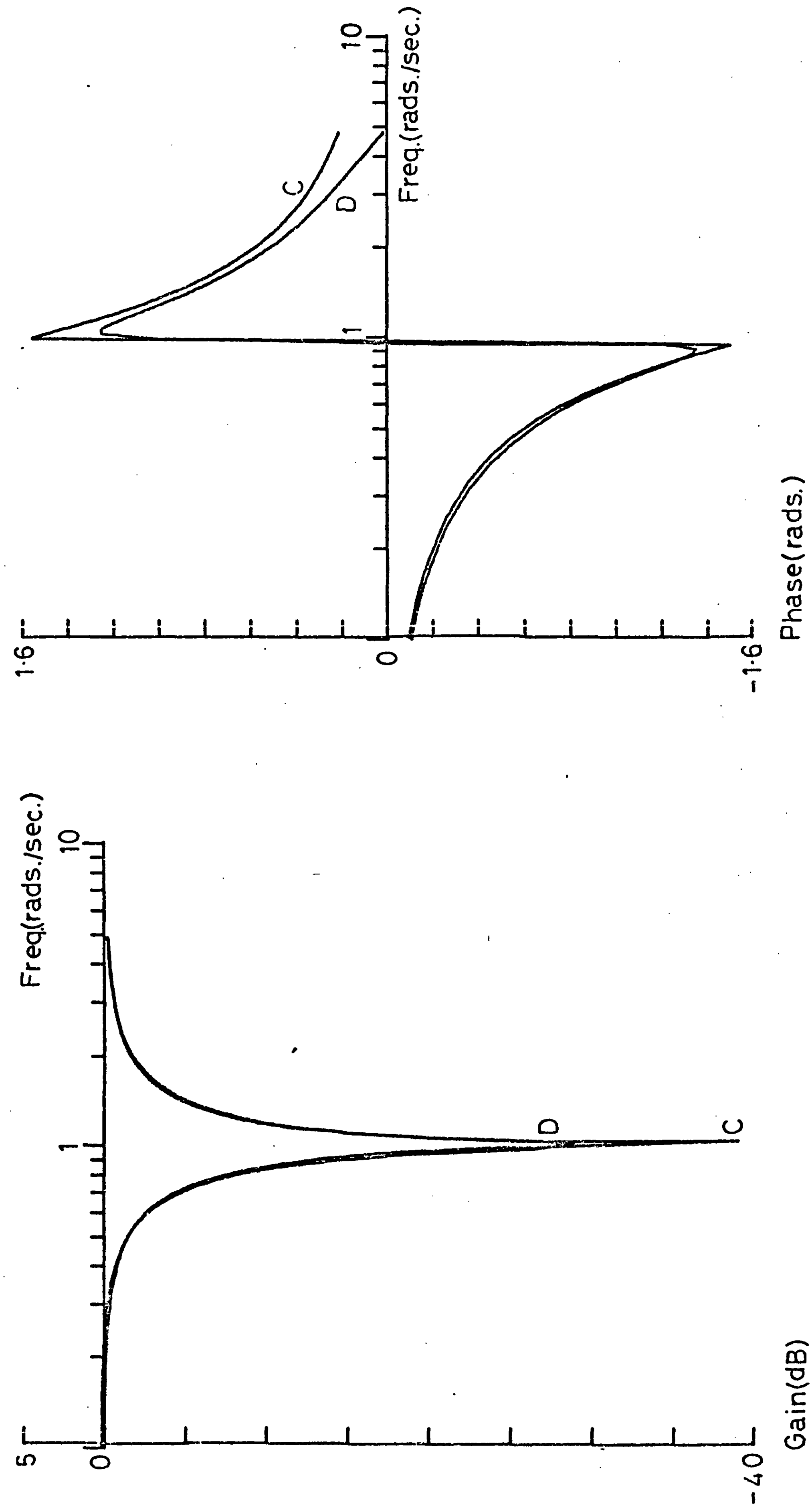


DESIGN RESULTS FOR A 2ND ORDER BAND-PASS FILTER FIGURE 5.5



THE RESULT FOR THE SIMULATION OF A 2ND ORDER BUTTERWORTH FILTER

FIGURE 5.6



THE RESULTS FOR A BAND-ELIMINATION FILTER FIGURE 5.7

6. A FREQUENCY SAMPLING APPROACH

6.1 The Method

The last chapter demonstrated the use of unit pulse response data in the production of digital simulation filters. This method used the desired frequency response as a starting point. Methods can be considered which use the frequency response as a design basis without the necessity of finding the unit pulse response in the process. Consider the transfer function:

$$H(Z^{-1}) = \frac{\sum_{i=o}^{I} p_i Z^{-i}}{1 - \sum_{k=1}^{K} q_k Z^{-k}} \qquad 6.1$$

whose frequency response will then be found by setting $Z^{-1} = e^{-j\omega T}$. Linear equations can be formed from Equation 6.1. Let the value of the frequency response at a particular frequency be X, then:

$$X = \sum_{i=o}^{I} p_i e^{-ji\omega T} + X \sum_{k=1}^{K} q_k e^{-jk\omega T} . \qquad 6.2$$

A set of such equations at various frequencies can be solved, for the values of p and q, if the number of equations m (say) is such that:

$$m \geq I + K + 1 . \qquad 6.3$$

The equation set can thus be written:

$$
\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ \vdots \\ \vdots \\ X_m \end{pmatrix}
=
\begin{pmatrix}
1 & e^{-j\omega_1 T} & e^{-j2\omega_1 T} --- e^{-jI\omega_1 T} & X_1 e^{-j\omega_1 T} & X_1 e^{-j2\omega_1 T} --- X_1 e^{-jK\omega_1 T} \\
1 & e^{-j\omega_2 T} & ------ & & \\
\vdots & & & & \\
\vdots & & & & \\
1 & e^{-j\omega_m T} & e^{-j2\omega_m T} & ---------------- & X_m e^{-jK\omega_m T}
\end{pmatrix}
\begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ p_I \\ q_1 \\ q_2 \\ \vdots \\ q_K \end{pmatrix}
\tag{6.4}
$$

The use of this approach is similar to methods involving Padé approximants (References 54, 56 and 59). Padé approximants are used to form polynomial rational fraction functions by equating them with the desired function as demonstrated above. For the design of simulation filters the values of 'X' are samples from the desired frequency response. The calculated values of 'p' and 'q' will then be such that the digital filter frequency response approximates the desired response. This method has the restriction that, unlike the unit pulse method, no guarantee of stability is inherent in the result. That is, the relationships expressed so far could equally apply to both stable and unstable results, whereas the considerations of Section 5.2 mean that the unit pulse method will produce stable results. In practice some unstable designs did result particularly in cases where the order of the simulation was increased when attempting to improve the quality of that simulation. In Chapter 5 a method was discussed of 'sensibly' relating the magnitude and phase responses with respect to the Z^{-1} plane frequency variable. The algorithm described in this section can be extended to include the use of IDFT and Hilbert transform to give 'sensible' frequency responses. In practice algorithms were used which used both 'direct' (with no modification) and 'indirect' (with the phase response related via the Hilbert transform) methods. As with the 'unit pulse' method it was necessary to include checks in the 'indirect' algorithm such that phase equalising and unity magnitude roots were represented accurately. These checks were again based upon the use of a known 'sensible' response, namely the equivalent Bilinear Z Transform response. Figure 6.1 shows the flow diagram of the 'direct' algorithm and Figure 6.2 the flow diagram of the 'indirect' algorithm.

6.2 Examples

The examples for the 'frequency sampling' design method are shown in Figures 6.3 to 6.7. The figures show graphs of the magnitude and phase responses. The curves on the graphs are marked C (for the continuous response) and D (for the digital response). In all examples the sampling frequency is 10 rads./sec.

(i) Figure 6.3 shows the results for a design where the continuous original is a fourth order Chebyshev filter. The filter is low-pass of 1 rad./sec. cut-off frequency and 3dB ripple. The results of the transformation design methods for this filter are shown in Figures 2.1 to 2.4. The 'zero insertion' method result is shown in Figure 4.2 and the 'unit pulse response' method is shown in Figure 5.2. The digital filter designs in this example were found by the 'indirect' method (that is, with Hilbert transform related phase). The digital responses are labelled:

D_1 for the minimum phase version;

D_2 and D_3 versions of the response with different zero invertions.

(ii) Figures 6.4 and 6.5 show the results for a second order band-pass filter. The transfer function of the continuous filter is:

$$H(s) = \frac{s}{s^2 + 0.7s + 1} \cdot \qquad 6.5$$

Figure 5.5 shows the results for the Standard Z Transform and the 'unit pulse response' method. Figure 6.4 shows the results of using the 'direct' frequency sampling approach. In Figure 6.4 the curves are designated:

D_1 for a second order simulation;

D_2 for a third order simulation.

Figure 6.5 demonstrates the results for the 'indirect' frequency sampling method. Here the curves are:

D_1 for the minimum phase version;

D_2 for the zero inverted version.

(iii) Figure 6.6 gives the results for a third order high-pass Butterworth filter, cut-off frequency 1 rad./sec. In this example the curves are:

D_1 designed by the 'direct' method;

D_2 designed by the 'indirect' method.

(iv) Figure 6.7 gives the results of designs using a second order band-elimination filter as the continuous original. The transfer function of the original filter is:

$$H(s) = \frac{s^2 + 1}{s^2 + s + 1} \quad . \qquad 6.6$$

Figure 5.7 shows the results achieved by the 'unit pulse response' design method. In Figure 6.7 the curves are denoted:

D_1 for the 'direct design;

D_2 for the 'indirect' design.

These results demonstrate the accuracy that can be obtained by use of the frequency sampling design method. The effects of using both the 'direct' and 'indirect' approaches are clearly seen. For the 'indirect' method the ramifications of using Hilbert transform related phase responses are shown. In short, the 'indirect' method produces very accurate magnitude response simulation, whereas the 'direct' method effects a compromise between phase and magnitude accuracy. A detailed discussion of these results, compared with the other simulation design methods, is to be found in Chapter 7.

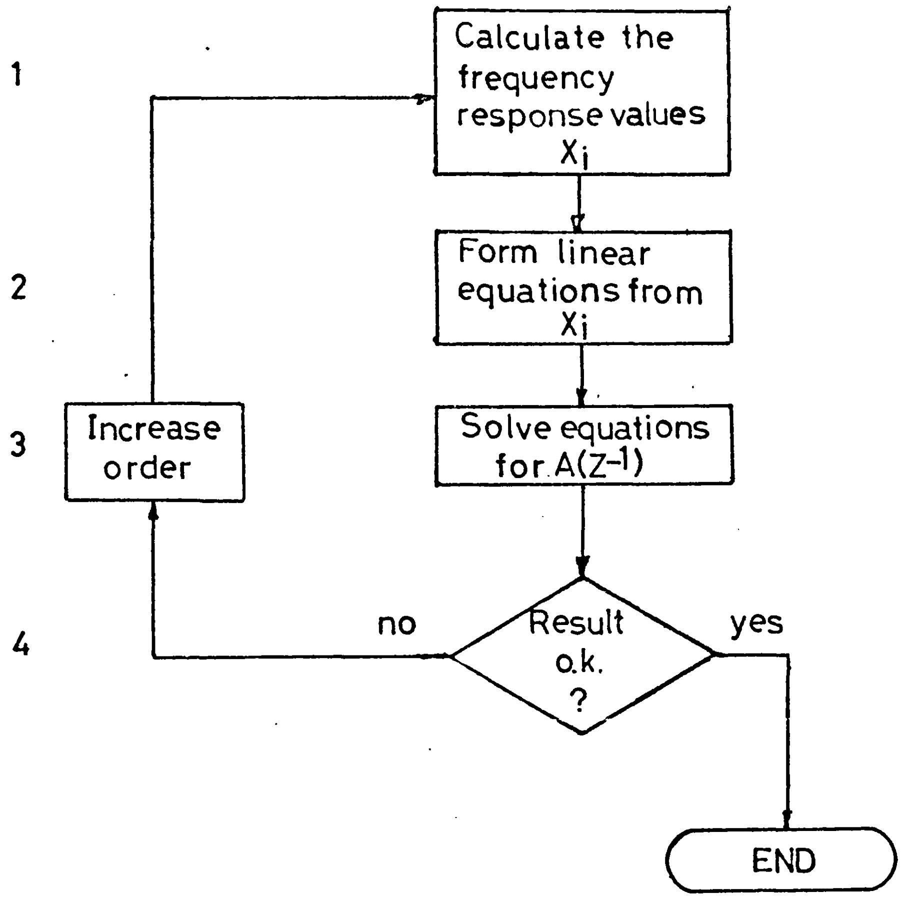


Fig.6.1. The 'Direct' Frequency Sampling Method.

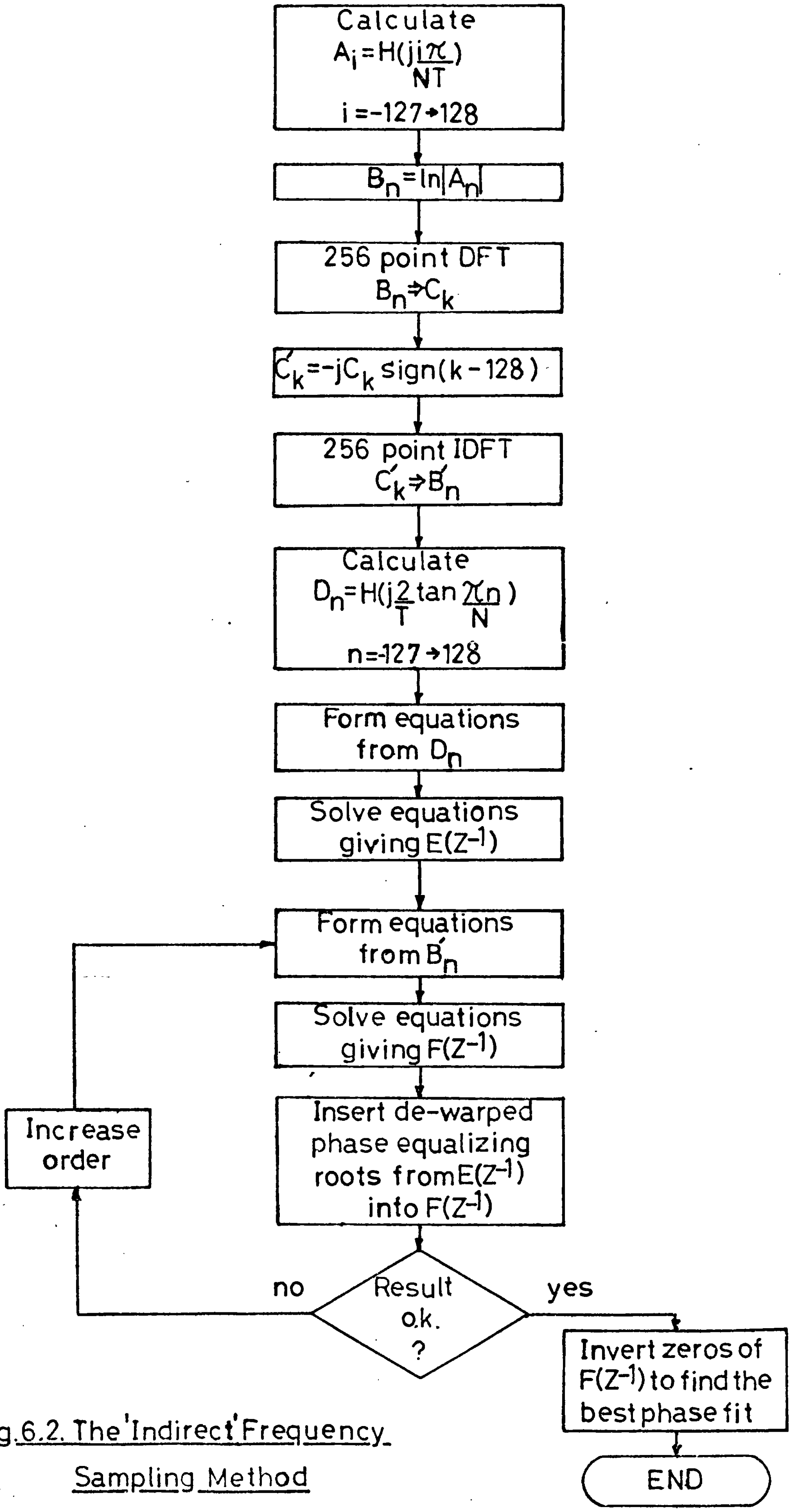
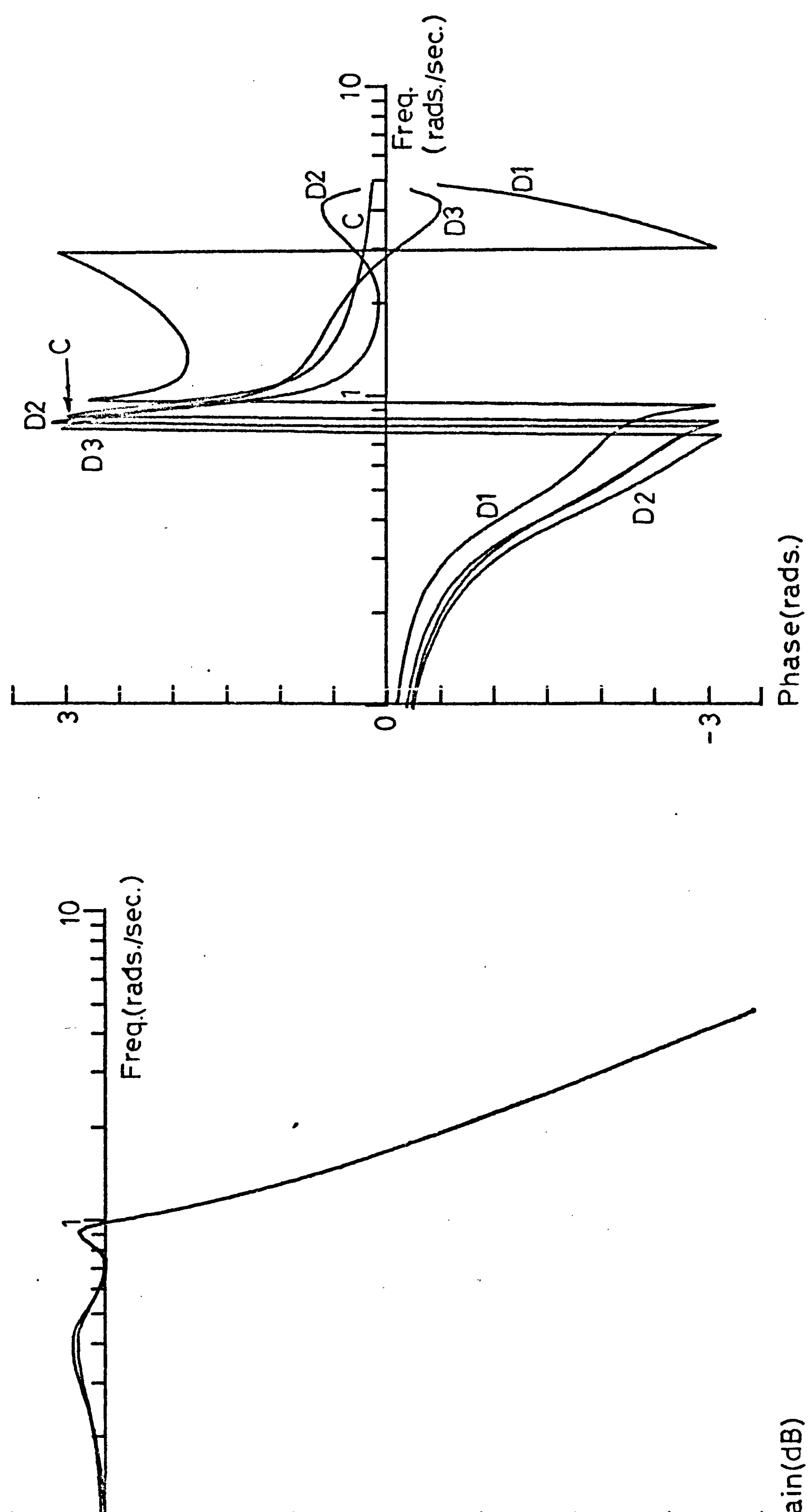
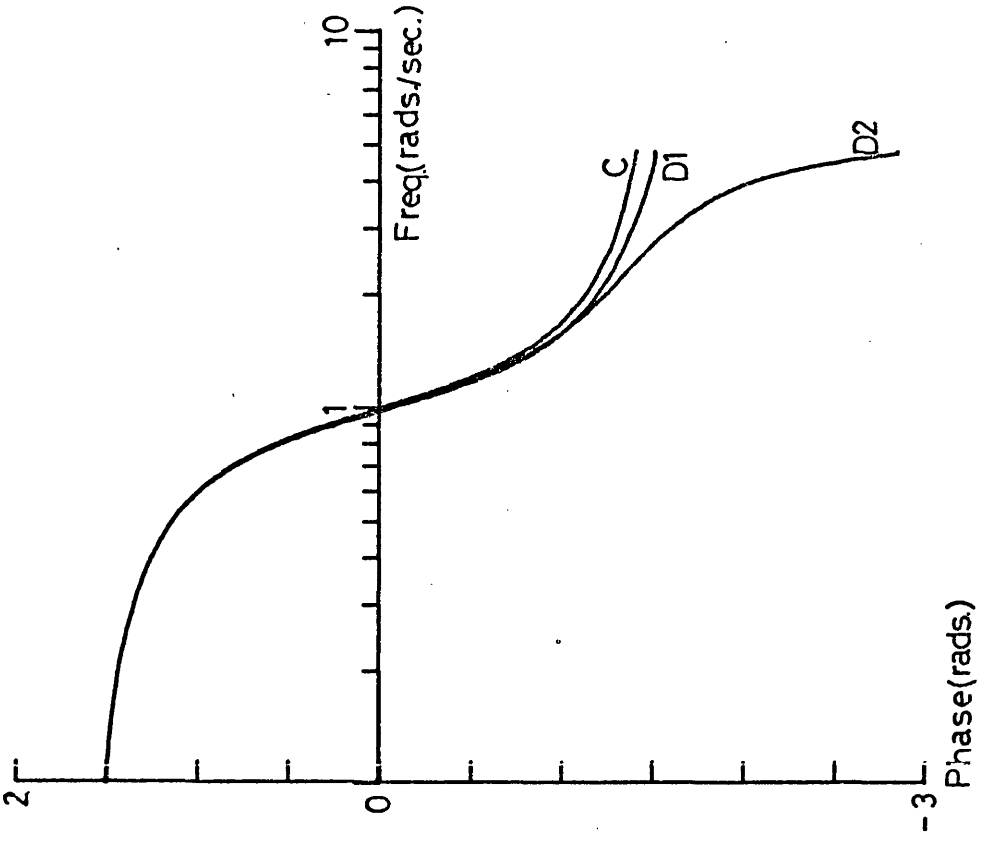
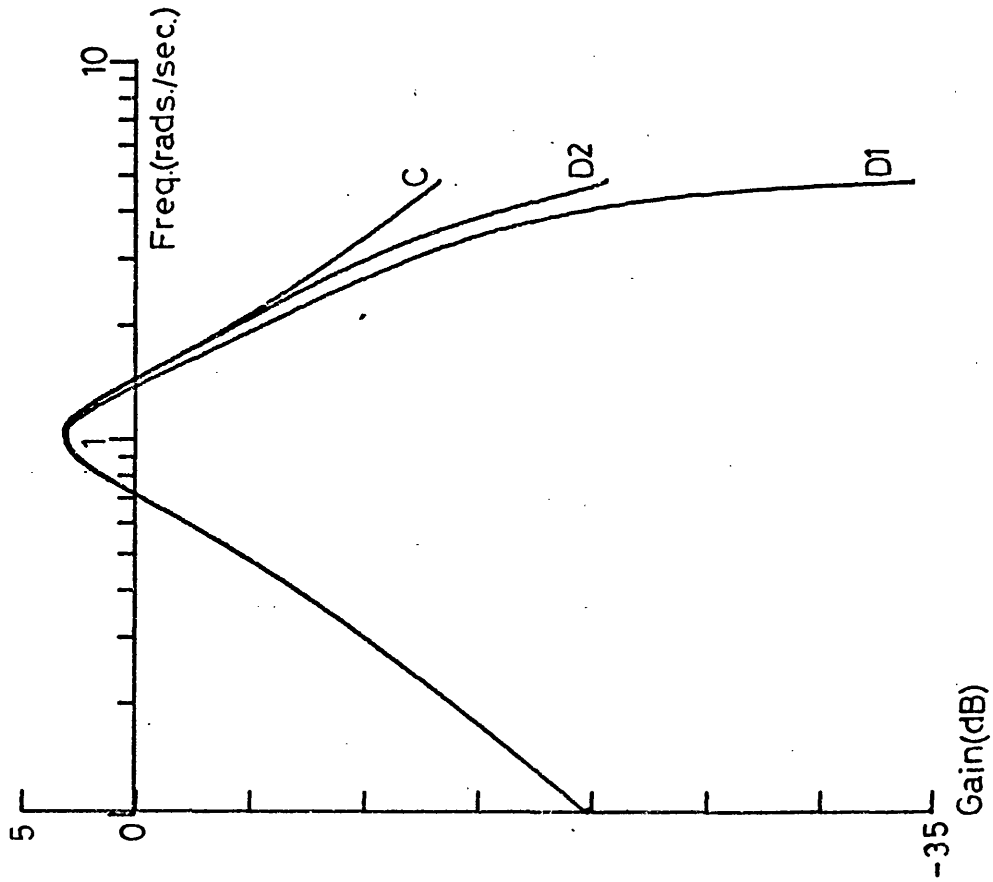


Fig.6.2. The 'Indirect' Frequency Sampling Method



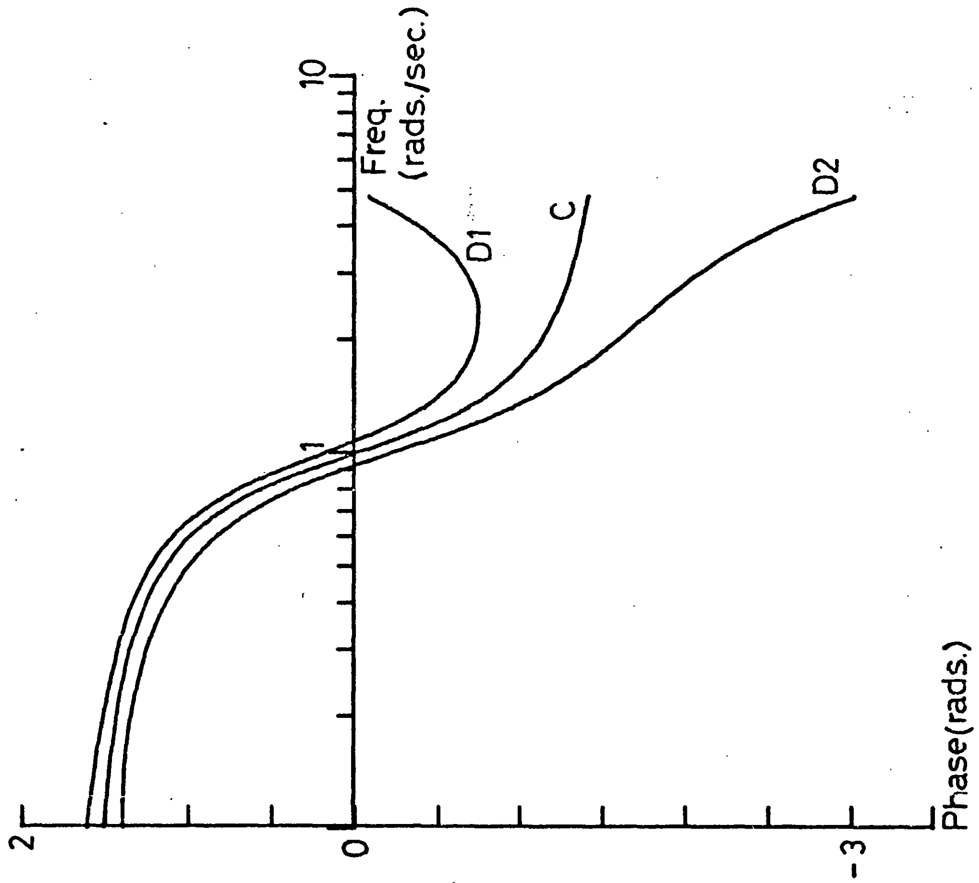
THE RESULTS FOR THE 4TH ORDER CHEBYSHEV FILTER FIGURE 6.3

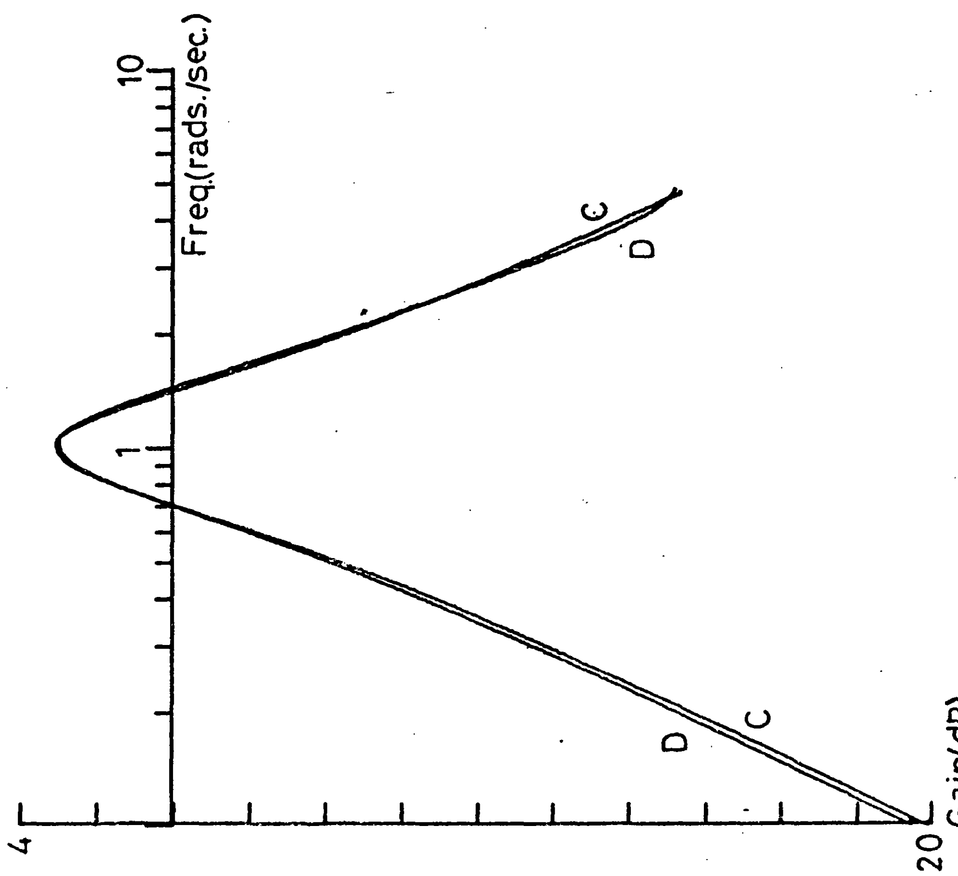




THE 'DIRECT' DESIGN METHOD RESULTS FOR THE 2ND ORDER BAND-PASS FILTER

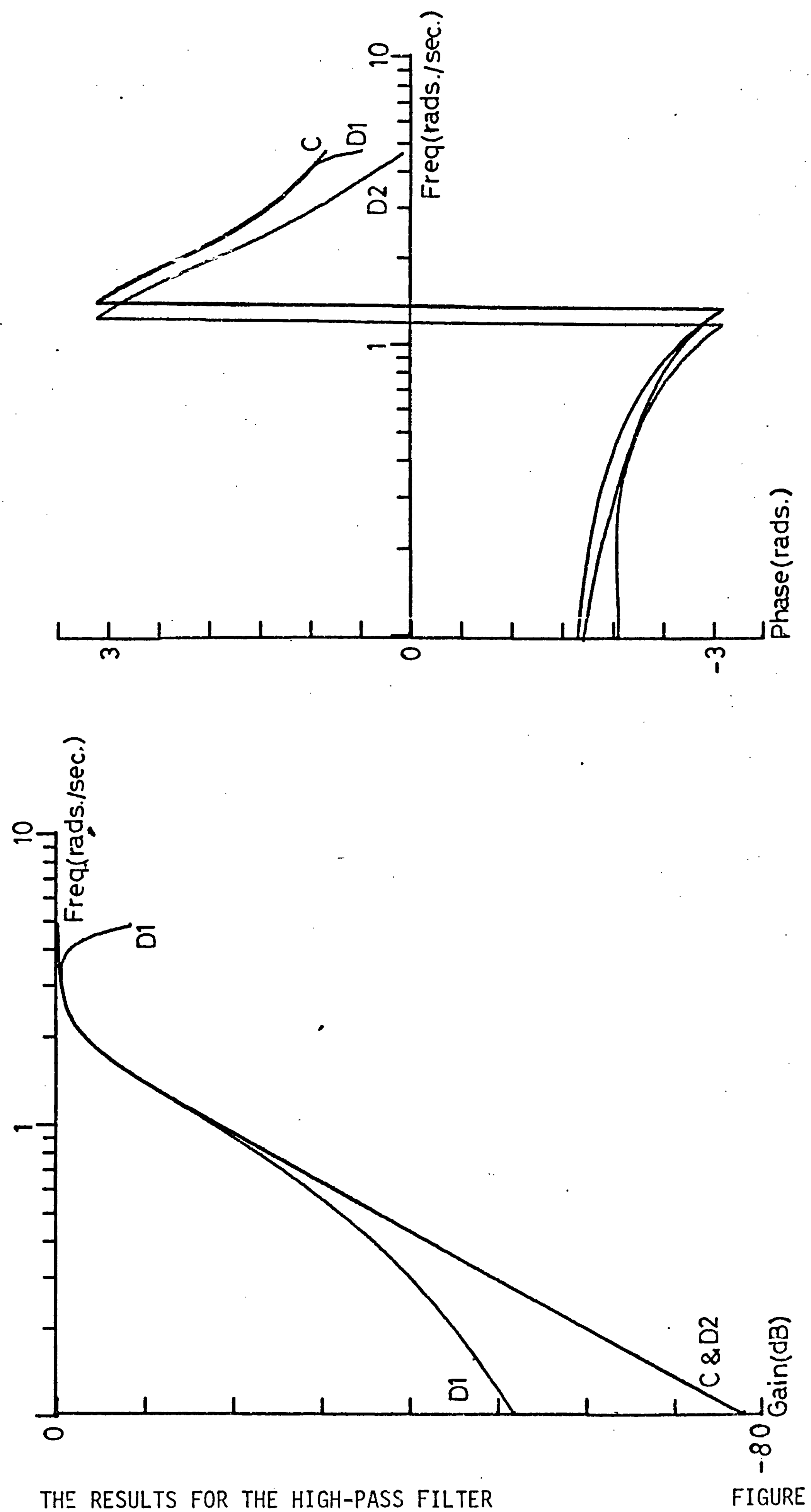
FIGURE 6.4



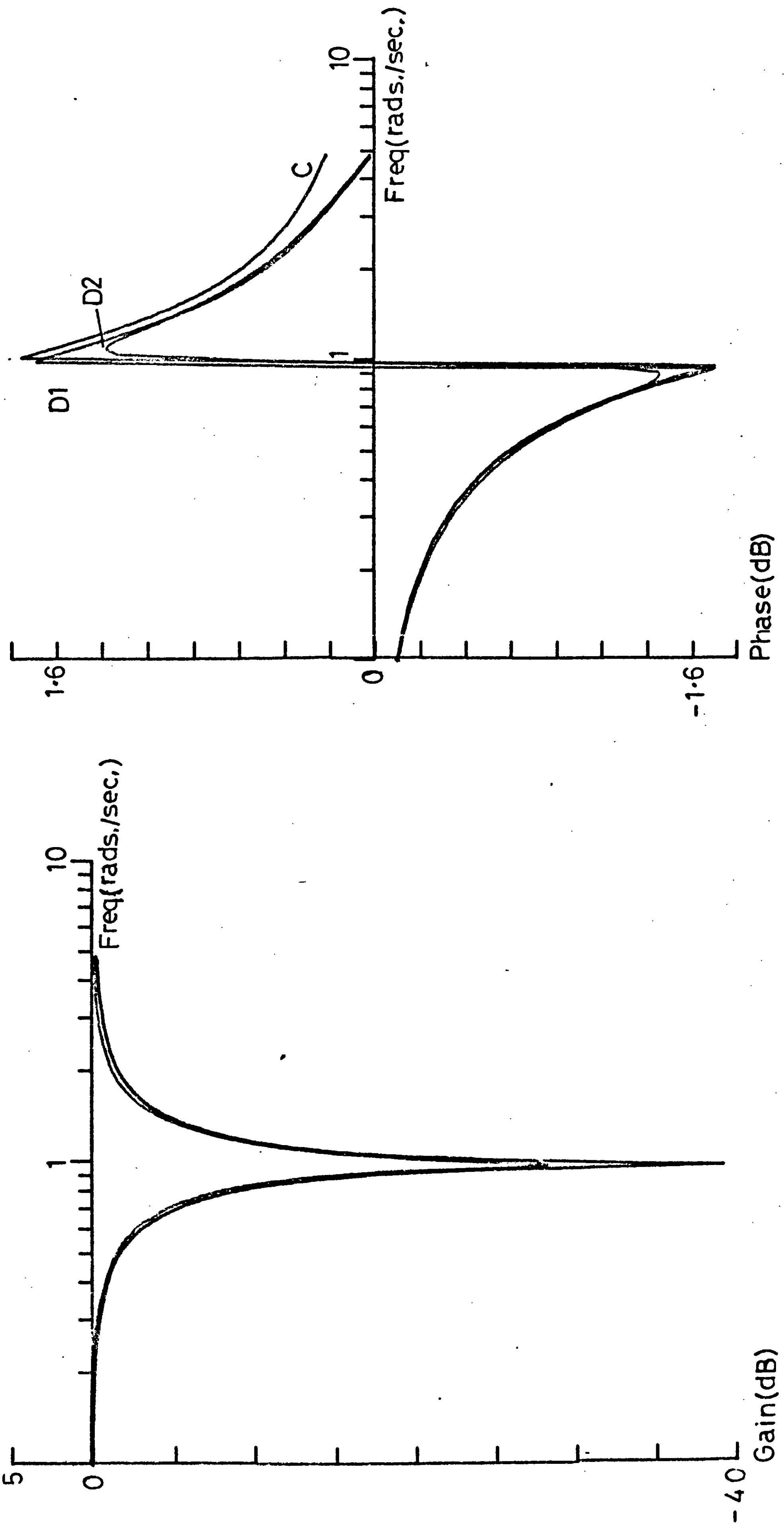


THE 'INDIRECT' DESIGN METHOD RESULTS FOR THE 2ND ORDER BAND-PASS FILTER

FIGURE 6.5



THE RESULTS FOR THE HIGH-PASS FILTER FIGURE 6.6



THE RESULTS FOR THE BAND-ELIMINATION FILTER FIGURE 6.7

7. DISCUSSION AND COMPARISON OF SIMULATION FILTER DESIGN METHODS

The previous three chapters have described and analysed different methods of designing digital filters. The differences between the methods are in the bases used to effect the design. The bases used were:

(i) the 's' plane transfer function;

(ii) the unit pulse response;

(iii) the desired frequency response.

The object of this study has been to produce digital filters which accurately simulate continuous frequency responses over the Nyquist range. The study has established the feasibility of designing digital simulation filters. The assessment of feasibility is performed by observing the accuracy of the results. The accuracy has only been measured by graphical comparison, but this is a sufficient basis to establish the feasibility of the methods. In a practical simulation environment however the accuracy must be assessed by quantitative means, this problem will be discussed later.

7.1 Discussion of Results

The examples of filter design used in the previous chapters were selected to illustrate particular points with respect to the design methods. Some types of response were used as examples for all design methods, to give a comparison facility. Other examples were used to illustrate points with respect to the individual methods.

7.1.1 Overall Comparisons

The example of the fourth order low-pass Chebyshev filter was used for all design methods. The results are shown in the following figures:

Figures 2.1 to 2.4 - transformation designs;
Figure 4.2 - the 'zero insertion' design;
Figure 5.2 - the 'unit pulse' design;
Figure 6.3 - the 'frequency sampling' design.

This example was used to check the capability of the design methods to reproduce a known good simulation. The Standard Z Transform result (Figure 2.1) shows that the transformation methods can produce an adequate simulation in this case. The results for the other three design methods show that good magnitude response simulations occur in all cases. The phase responses of the 'unit pulse' and 'frequency sampling' methods are not as good as for the 'zero insert' result. This is due to the use of the Hilbert transform to produce a magnitude related phase response in the design process. The best magnitude response is obtained using the 'unit pulse' method but the best overall result is found by the zero insertion method. Figure 6.3 also shows the effects, on the phase response, of zero inversion selection after use of the Hilbert transform rationalisation procedure.

Examples of low-pass filters with the cut-off frequency high in the Nyquist range can be used to check the ability of the design methods to produce simulations where the transformation methods fail. The example of the third order Butterworth filter with cut-off frequency 4 rads./sec. is just such an example.

Figures 2.5 to 2.8 show the transformation results;

Figures 5.3 and 5.4 show 'unit pulse' method results.

In this case the three transformation results are a total failure. The 'unit pulse response' design method however produces a good simulation of the magnitude response. The phase response simulation is slightly better in Figure 5.4 than in Figure 5.3. Both 'unit pulse' examples required the order of the simulation to be higher than the order of the original filter. Figures 4.3, 4.4 and 4.5 illustrate a similar example in terms of the 'zero insert' method. Figure 4.3 shows the poor results in the case of the transformation design. Figure 4.6 shows the responses for the final 'zero insertion' design, which can be seen to be quite a good simulation.

Two further examples are used for comparison of the 'unit pulse' and 'frequency sampling' methods. Firstly the case of the second order band-pass filter seen in Figures 5.5, 6.4 and 6.5.

Figure 5.5 shows the Standard Z Transform and 'unit pulse response' results.

Figure 6.4 shows the results for the 'direct' approach to the 'frequency sampling' method, and Figure 6.5 shows the 'indirect' approach result.

The Standard Z transform design does not result in a good simulation. For the 'unit pulse response' designs a third order simulation is not adequate but the fourth order simulation is reasonable. This example is illustrative of the problems that occur when the original filter phase shift is approximately equal to an odd multiple of $\pi/2$ radians at the Nyquist frequency. These problems are discussed in Chapters 3 and 4. In such a case a zero is required at $Z^{-1} = -1$ if the digital Nyquist frequency phase is to be close to the required phase. The 'direct frequency sampling' designs illustrate this problem. The results shown in Figure 6.4 give simulations for which the phase is much closer to the requirement than for the 'indirect' method shown in Figure 6.5. However the closer phase response results in magnitude response errors, especially in the case of curve D1 in Figure 6.4 where a zero has been used at $Z^{-1} = -1$. Figure 6.5 also illustrates the problems of choosing the zero inversions in such a case, by showing the minimum and maximum phase versions of the design.

Figures 5.7 and 6.7 give the responses for a second order band elimination filter. This example is important in that it illustrates the problems that can occur when the digital filter is required to have a root on the unit circle in the Z^{-1} plane. The band elimination filter continuous transfer function has zeros on the imaginary axis in the s plane, thus requiring zeros on the unit circle in the Z^{-1} plane. The problems with any method which is based upon samples of the required frequency response is that the samples may not accurately represent the depth of the magnitude response in the vicinity of the zeros. In addition if the Hilbert transform phase relating method is used it is necessary to obtain the logarithm of the magnitude, which is not practical where the magnitude approaches zero. In the figures for this example this is illustrated by the reduction in the depth of the notch.

7.1.2 The 'Zero Insert' Method

The examples shown in Chapter 4 show that the method of using the Matched Z Transform with extra zeros placed in the Z^{-1} plane, is viable for producing digital simulation filters. Two methods were used for finding the positions of the inserted zeros:

(i) by calculation;

(ii) by root searching followed by iterative refinement of the zero positions.

These methods were adequate for demonstration but are not practical for use in an automated simulation-filter design method. For practical circumstances a better method would be to calculate the zero positions by use of a linear equation scheme as used in Chapters 5 and 6. The fitting of linear equations is discussed later in this section.

The results of Chapter 2 show that the transformation methods can often produce reasonable frequency response simulations. The advantage of the zero insertion method is that it uses a transformation method to provide a firm basis for the design algorithm. The method, of course, requires an 's' plane transfer function as the filter specification. Where this type of specification is used very useful simulations result.

7.1.3 The 'Unit Pulse Response' Method

The 'unit pulse response' method is based upon the approximation of the IZT by the IDFT. The ramifications of this approximation have already been discussed. The method only works where the frequency response used to generate the unit pulse response is approximately rational in terms of the Z^{-1} variable. The practical algorithm approached rationality by relating the phase response to the magnitude response by using the Hilbert transform. This relationship means that the magnitude responses are very accurately simulated, but that the phase responses are not necessarily so accurately simulated. These points can be seen in the examples. Reduction of the phase response error could possibly be effected by the use of complementary pole-zero pairs. The frequency responses also show that the quality of the simulation can often be improved by increasing the order of the simulation. In the described algorithm this increase of order is performed with respect to an error criterion. Error criteria will be discussed in more detail later in this chapter.

Another important feature of the 'unit pulse response' method is its guaranteed stability. That is, if the conditions of Section 5.2 are met then the unit pulse response must be convergent and the resultant filter stable. This condition is not guaranteed by the 'frequency sampling' method.

The studies reported here also produced a very interesting side effect. In Section 5.3 the feasibility of using the unit pulse response was established by employing a frequency response that is equivalent to the frequency response of a Bilinear Z Transform digital filter. This procedure provides a method for finding the transfer function of a continuous system whose frequency response is known. This method is very similar to the Wiener-Lee transforms.

7.1.4 The 'Frequency Sampling' Method

Two methods of using the frequency response itself as a design basis are investigated. Firstly the 'direct' method, where the desired response is used directly in the linear equations. Secondly the 'indirect' method, where the phase response used in the design algorithm is related to the magnitude response by the Hilbert transform.

The 'indirect' algorithm is similar to the 'unit pulse response' method in its effects. This is evident in the accurate modelling of the magnitude of the original frequency response.

The 'direct' method shows results which are different from those of the 'indirect' method. These differences are manifested in the results by more accurate modelling of the phase response at the cost of the magnitude response accuracy. This point is illustrated by Figures 6.4 and 6.5.

The arguments about Hilbert transformation and increasing the simulation order are the same as for the 'unit pulse response' method. However the 'frequency sampling' method carries none of the guarantees of stability that are inherent in the 'unit pulse response' method. In practice unstable results have occurred in some cases when the order of the simulation was increased. The instabilities tend to occur more often with the 'direct' method. This is probably due to the fact that the phase response required could be grossly unrealisable (see Chapter 3).

Overall the results show that the 'frequency sampling' method can produce good simulations. As with the 'unit pulse response' method the use of the Hilbert transform related phase response gives very good magnitude response simulation.

7.2 Accuracy and Fitting Considerations

All of the algorithms discussed have required some method of assessing the accuracy of simulation and all have used methods of fitting equations to data.

7.2.1 Accuracy

The accuracy of simulation has been used to aid the process of digital filter design. For example, in the 'unit pulse response' and 'frequency sampling' methods accuracy considerations were used to decide upon the advisability of increasing the order of simulation. As has been previously stated the accuracy checks used in the algorithms were suitable for testing purposes only. In the real world a quantitative measure of accuracy of the frequency response is required.
Such an assessment could be implemented by means of least square or minimum deviation methods. No problems are envisaged in using these types of method in the algorithms as described. The question that does remain is, which method of assessment is to be used? It is difficult to state a definite answer to this question, and to do so could well restrict the usefulness of the overall algorithm. It is best to be flexible in this matter and allow a variety of methods of accuracy evaluation. This approach will allow the user the freedom to define the quality of his simulation.

7.2.2 Equation Fitting

The algorithms as described all involved the calculation of equation coefficients, that is, a 'fit' was obtained of the equation to the data. This 'fit' was found by formulating a set of linear equations and then calculating the coefficients of those equations. That is:

$$y_i = \sum_{j=1}^{n} x_{i,j} \, c_j \qquad 7.1$$

where the values of y_i and $x_{i,j}$ are known and it is required to calculate the coefficients c_j. If a set of 'm' independent equations are formed a solution is possible if $m \geqslant n$. The 'calculation' method used for zero insertion found the solution when $m = n$. Early methods developed for the 'unit pulse response' and 'frequency sampling' algorithms also used $m = n$. The solution arrived at in these circumstances provides a fit which gives exact equality of the responses at 'm' points. The exact fit at a certain number of points can mean considerable deviation of the response between these points. This was found to be a problem in the 'unit pulse response' and 'frequency sampling' methods. In the 'direct frequency sampling' method the problems were manifested in both inaccurate results and often unstable filter designs. For these reasons it is better that $m > n$ (and preferably $m >> n$). Such a set of equations are known as overdetermined, that is, more equations are used than are absolutely necessary (the problem is overdetermined). Such overdetermined equations were used in the algorithms implemented for the 'unit pulse response' and 'frequency sampling' methods. Now, when calculation is performed using a set of overdetermined equations the solution is found by minimisation of an error criterion. That is, the current solution is tested against the data and accepted if the error criterion is minimised. If the error criterion is not minimised then a new solution is found and tested. The scheme for the solution of overdetermined equations used by the author in the design algorithms already described was that reported by Bartels and Golub (Refs. 2 & 3). This method solves the equations with respect to a maximum deviation error criterion (known as 'mini-max'). Another method that has been observed in the literature is a least square error method reported by Evans and Fischl (Ref. 17).

7.3 Conclusion

The previous results and discussions have shown that digital filters can be designed to model the frequency responses of continuous filters. The quality of the simulations can be seen from the examples. Particularly good simulations of the magnitude response are possible when the phase response of the digital filter is related to the magnitude response by the use of the Hilbert transform. It is felt that in these cases better phase response accuracy could be achieved by the use of phase equalisation methods. Where the Hilbert transform is not used the 'frequency sampling' method attempts to simulate both magnitude and phase responses often with reasonable success.

Four methods have been used for digital filter design.

7.3.1 The Zero Insert Method

Using the 'zero insert' method filters are designed from 's' plane transfer functions by use of the Matched Z Transform and extra zeros are inserted into the Z^{-1} plane to provide the result. This method is useful in that it makes use of the good basis of design provided by s to Z^{-1} plane transformation. The algorithms for zero insertion described in the text require modification for them to be useful in a system simulation scheme. Suggestions for modifications are made in the text.

7.3.2 The Unit Pulse Response Method

This method designs filters by forming linear equations from a unit pulse response. The unit pulse response is found from the desired frequency response and that frequency response is formed by using the Hilbert transform phase relationship. This method produces very good simulation of the magnitude response of the original filter. However, the results achieved by this method are similar to those found by using the 'indirect frequency sampling' method.

7.3.3 The 'Indirect' Frequency Sampling Method

In this case the designs are effected by forming linear equations from the Hilbert transform phase related frequency response. The results are similar to the 'unit pulse response' method but there is no need to find the unit pulse response of the filter. Therefore design by the 'frequency sampling' method requires less computational effort.

7.3.4 The 'Direct' Frequency Sampling Method

This algorithm also uses linear equations formed from the desired frequency response. In the 'direct' method however the phase response is not related by the Hilbert transform but is the actual phase response desired. This means that the resultant digital frequency response is a 'trade-off' between magnitude and phase accuracy. The method has the disadvantage that unstable designs sometimes result when the order of the design is increased in an attempt to improve the quality of the simulation.

8. FUNCTION GENERATION

In previous chapters the design of digital filters has been investigated with respect to the simulation of the continuous system frequency responses. The accuracy of simulation has been judged by comparison of the continuous and digital frequency responses. If the hypothesis of frequency response simulation is to be accepted then it must also apply to the function generating elements in a system. That is, the spectrum of the simulated function generator should be as close as possible to the spectrum of the element in the original system over the Nyquist range. Only in this way can the alias errors, involved in sampling, be avoided.

It is the intention of this chapter to investigate the simulation of function generators, both bandlimited and non-bandlimited. The problems involved in generating non-bandlimited functions are virtually ignored by the literature (e.g. References 19, 27 and 48). In the literature the alias problems are overcome by simply increasing the sampling rate until the error becomes negligible. However, for economic reasons, it is desirable to keep the sampling rate as low as possible, hence this study. It must be kept in mind of course that the highest frequency of interest in a system must be accurately represented. This factor will determine the minimum sampling rate to be used.

Most driving functions can be placed in one of three categories, namely periodic, aperiodic and random functions. The following study will involve consideration of signals in each of these categories.

8.1 Periodic Functions

8.1.1 Sinusoidal Functions

The production of sinusoidal functions has none of the limitations due to alias errors as, by definition, such functions are band-limited. Several methods may be considered for the generation of sampled sinusoids:

(a) By direct calculation.

Most computer systems have facilities for the calculation of values of sinusoidal functions. Therefore for:

$$x(nT) = \sin(n\omega T) , \qquad 8.1$$

it is simply necessary to use the value $n\omega T$ and thus calculate $x(nT)$ for a sinusoid of frequency ω. In most computer systems an increase in calculation speed will result for large values of n if the value:

$$\theta = n\omega T \text{ modulo } 2\pi , \qquad 8.2$$

is used as the angle for the sinusoid. The disadvantage of direct calculation of sinusoidal values is that the operation is often time consuming when compared to many typical calculations.

(b) Digital oscillators

Bogner (Ref. 4) and Saltzberg (Ref. 49), have investigated the use of digital filters with unity magnitude Z^{-1} plane poles (i.e. lying on the unit circle) to generate sinusoidal functions. For a frequency ω in the Nyquist range the transfer function of the digital oscillator is:

$$G(Z^{-1}) = \frac{1}{(Z^{-1} - e^{j\omega T})(Z^{-1} - e^{-j\omega T})} \quad . \qquad 8.3$$

For a complex signal of frequency ω the transfer function becomes:

$$H(Z^{-1}) \Big| = \frac{1}{1 - e^{-j\omega T}Z^{-1}} \quad . \qquad 8.4$$

Bogner illustrates the point that Equation 8.4, when applied, generates both sine and cosine waveforms for no extra computational cost over Equation 8.3. The schematic of the digital oscillator is shown in Figure 8.1, where $x(nT)$ represents a starting sequence and $y(nT)$ the complex sinusoid, that is:

$$y(nT) = \cos(n\omega T) + j \sin(n\omega T). \qquad 8.5$$

The complex oscillator may be started at any phase angle by choice of the sequence $x(nT)$, if:

$$x(o) = \cos\theta + j\sin\theta \qquad 8.6$$

$$x(nT) = o, \text{ for all } n \neq o$$

$$y(nT) = \cos(n\omega T + \theta) + j \sin(n\omega T + \theta). \qquad 8.7$$

The advantage of this system is that it is computationally very simple. Practical computers, however, have a finite word length. The finite word length can mean that the pole of Equation 8.4 may not lie exactly on the unit circle. In such a circumstance the oscillator output will decay or grow exponentially at a rate dependent upon the radial distance of the pole from the unit circle. This problem can be overcome by ensuring that the feedback value $p(nT)$, (see Figure 8.1) is of unit modulus. That is, the modified feedback value becomes:

$$p'(nT) = \frac{p(nT)}{|p(nT)|} \qquad 8.8$$

In practical computation terms the calculation of $|p(nT)|$ will include the calculation of a square root, which can be time consuming. Consider the case where the modulus of the pole position is in error by δ, then:

$$\left|p(nT)\right| = (1 + \delta)\left|y(nT-T).\right| \qquad 8.9$$

Now consider omitting the abovementioned square root operation, then:

$$p'(nT) = \frac{p(nT)}{|p(nT)|^2} \qquad 8.10$$

that is:

$$|p'(nT)| = \frac{|y(nT-T)|}{(1 + \delta)} , \qquad 8.11$$

thus:

$$|y(nT)| = \frac{|y(nT-T)|}{(1 + \delta)} . \qquad 8.12$$

By further similar analysis the modulus of oscillator output sequence can be seen to vary between unity and $1/(1+\delta)$ on alternate samples. Now, the value of δ can be expected to be small and so the extent of the variation can be expected to be small. Therefore, if a machine of reasonable word length is to be used (i.e. if δ is small) then the square root operation can be ignored. Thus stability is guaranteed at low computational cost.

(c) 'Look-up' tables

A 'look-up' table is simply a table of values which is stored in the computer and referred to as necessary. For the generation of sinusoidal functions the values in the table would be sinusoidally related. The frequency of the resultant sinusoid is dependent upon the intervals at which the table is sampled. The 'look-up' table method has the restriction that only a finite number of frequencies can be generated. For a table of 'n' values, say, the values representing one cycle of a sinusoid, then the only frequencies that can be generated are defined by:

$$\omega_i = \frac{i\omega_s}{n}, \qquad 8.13$$

where 'i' is integer and $i \leqslant \frac{n}{2}$.

The size of the table can be reduced at the cost of a small amount of computer time when it is remembered that:

$$\sin(\theta) = \sin(\pi-\theta) = -\sin(\pi+\theta) = -\sin(2\pi-\theta), \qquad 8.14$$

where $0 \leqslant \theta \leqslant \pi/2$, thus reducing the size of the table by 75%.

8.1.2 Non-Sinusoidal Periodic Functions

Non-sinusoidal periodic functions can be constructed from sinusoidal functions by virtue of Fourier series theory. Using this approach the individual sinusoids could be generated by any of the methods described in Section 8.1.1. For a period τ the maximum number of sinusoids required will be the largest integer value 'n' such that:

$$n \leqslant \frac{\tau}{2T}. \qquad 8.15$$

The 'look-up' table approach can also be applied to non-sinusoidal functions. Look up tables would only be practical where $\frac{iT}{\tau}$ is an integer given that 'i' is integer.

Both of these methods become impractical for functions with very large periods. Certain commonly used functions do have very large periods. For example suppose that a 'k' bit pseudo-random sequence is to be generated and let the individual bits be τ seconds wide. Then the number of sinusoidal generators required would be:

$$n = \frac{(2^k - 1)\tau}{2T} \qquad 8.16$$

remembering that $\tau > T$ shows that this number of generators could be impractical. Another method may be considered for generating non-sinusoidal periodic functions. Consider the possibility of generating a sampled version of the non-bandlimited function and then correcting the resultant alias error by means of a digital filter. This technique is only applicable where no spurious signals are produced by the alias effect. To ensure the absence of spurious signals it is necessary that the sampling rate is such that the folded harmonics occur exactly at frequencies of existing harmonics in the Nyquist band. A condition for coincidence of harmonics is that the period:

$$\tau = nT \quad . \qquad 8.17$$

This condition is necessary but not sufficient to ensure coincidence for example in the case of square waves 'n' must be even, as only odd harmonics exist. For demonstrative purposes consider the simulation of a sampled square wave function. The envelope of the spectrum of a square wave function can be found by considering the Fourier analysis of the equivalent square pulse. For a square pulse g(t) of width τ and unit amplitude then:

$$G(\omega) = \frac{e^{-j\omega\tau}-1}{-j\omega} \quad . \qquad 8.18$$

Now consider a sampled version of g(t), g*(t) also allow a small time delay (α) to avoid problems due to samples being coincident with the transitions of the pulse. Then for a unit amplitude pulse:

$$g^*(t) = \delta(t - kT - \alpha), \qquad 8.19$$

where k = 1, 2, 3, 4 ---- m, m being integer and:

$$m \lessgtr \frac{\tau}{T} .$$

Therefore,

$$G^*(\omega) = \sum_{k=1}^{m} e^{-j\omega(kT+\alpha)} , \qquad 8.20$$

making α infinitesimal gives:

$$G^*(\omega) = \sum_{k=1}^{m} e^{-jk\omega T} . \qquad 8.21$$

Normalising Equations 8.18 and 8.21 such that the magnitude at zero frequency is unity gives:

$$G'(\omega) = \frac{e^{-j\omega\tau}-1}{-j\omega\tau} , \qquad 8.22$$

and:

$$G^{*\prime}(\omega) = \frac{\sum_{k=1}^{m} e^{-j\omega kT}}{m} . \qquad 8.23$$

Now, the signal to be generated is $g^*(t)$ and this signal is to be filtered such that the spectrum is that of $g(t)$ over the Nyquist band. The function of the filter required is therefore:

$$H(\omega) = \frac{G'(\omega)}{G^{*\prime}(\omega)} = \frac{m(e^{-j\omega\tau}-1)}{-j\omega\tau \sum_{k=1}^{m} e^{-jk\omega T}} . \qquad 8.24$$

The requirement that no spurious signals be produced means that:

$$m = \frac{\tau}{T} . \qquad 8.25$$

Therefore:

$$H(\omega) = \frac{(e^{-jm\omega T}-1)}{-j\omega T \sum_{k=1}^{m} e^{-jk\omega T}} \qquad 8.26$$

which becomes:

$$H(\omega) = \frac{e^{-j\omega T}-1}{-j\omega T\, e^{-j\omega T}} \;. \qquad 8.27$$

This response can be seen to have linear phase and be independent of m. Any linear phase version of the response will be adequate at the cost of a time delay. Using the methods of Chapter 6 a digital filter was designed with respect to the requirements of Equation 8.27. The frequency response of the resultant filter, compared with the magnitude of Equation 8.27 and a linear phase response, can be seen in Figure 8.2. The transfer function of the resultant filter is:

$$F(Z^{-1}) = \frac{0.08646 + 0.91315Z^{-1} + 0.4857Z^{-2}}{1 + 0.48532\, Z^{-1}} \;. \qquad 8.28$$

The result of passing square pulse functions through such a filter have been spectrum analysed and Table 8.1 shows that the filter does the job. The importance of this result is that it shows that only one design of digital filter is required for accurate simulation of any function based upon square pulses. The only restriction is that the pulse width of the function must be an integral number of sampling intervals.

A similar analysis of a triangular pulse signal gives a correction filter response:

$$H(\omega) = \frac{(1 - e^{-j\omega T})^2}{\omega^2 T^2 e^{-j\omega T}} \;, \qquad 8.29$$

which is of linear phase and independent of the number of samples per pulse.

It is interesting to note that the response of Equation 8.29 is equivalent to the product of two responses of Equation 8.27, when linear phase considerations are disregarded.

Spectral analyses of other types of periodic signals has shown that the independence, with respect to the number of samples per pulse, of the required correction response is not a universal phenomenon. However, the principle of the method of using correction filters is established by the square pulse example. The value of this method is probably most evident when signals of large period are involved. This point is demonstrated by the fact that the data used to produce Table 8.1 was the spectrum analysis of part of a 12 bit pseudo-random sequence. Such a sequence has a very large period.

8.2 Aperiodic Functions

The problems of the generation of aperiodic functions are similar to those of periodic functions where the aperiodic cases are time limited. This is demonstrated in Section 8.1 where pulse functions were used to form the equations for the spectral correction. For other types of aperiodic function the problems of spectral invariance are more difficult. The simulation of this class of functions is probably more akin to digital filter design, in that some approximation to the spectrum will be necessary. A possible approach to this problem is the use of a digital filter whose unit pulse response is the model of the required function. No direct work has been done in this area due to lack of time. However it is necessary to identify the problem which is stated above together with a possible approach.

8.3 Random and Pseudo-Random Functions

Pseudo-random binary sequences are widely used for the analysis of systems. The work reported in Section 8.1 deals with the simulation of such sequences by the realisation of a universal approach to binary pulse type signals.

Random functions are dealt with in the simulation literature (References 19, 52 and 63). This work seems applicable but no detailed study has been done. It can be said that there seems to be no basic problem with the generation of random functions. Gaussian and other probability distributions are capable of computer generation. The engineering requirement is for both white and band-limited noise. Band-limited random functions can be produced from white noise functions and the use of digital filters. This area obviously requires detailed study to see if the above assumptions and hypotheses are reasonable.

Harmonic	Spectral Magnitude			
	Required	Measured	Filter Gain Required	Corrected Measurement
0	$\pi/2$	$\pi/2$	1	$\pi/2$
1	1	.9985	.9985	.9998
3	.3333	.3382	.9856	.3328
5	.2	.2083	.9603	.1993
7	.1429	.1547	.9231	.1422
9	.1111	.1271	.8749	.1107
11	.09091	.1114	.8167	.09093
13	.07692	.1026	.7498	.07686
15	.06667	.09865	.6758	.06678

SPECTRAL MEASUREMENTS FOR A 16 SAMPLE/BIT SQUARE-WAVE FUNCTION

TABLE 8.1

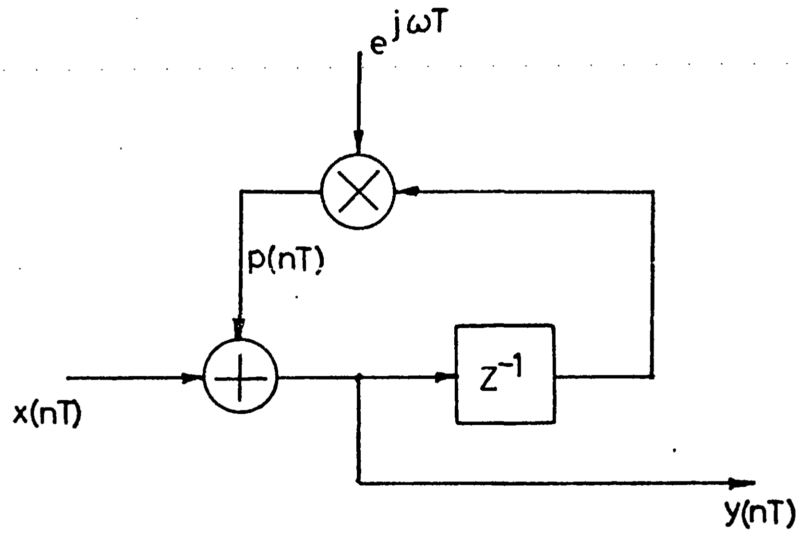
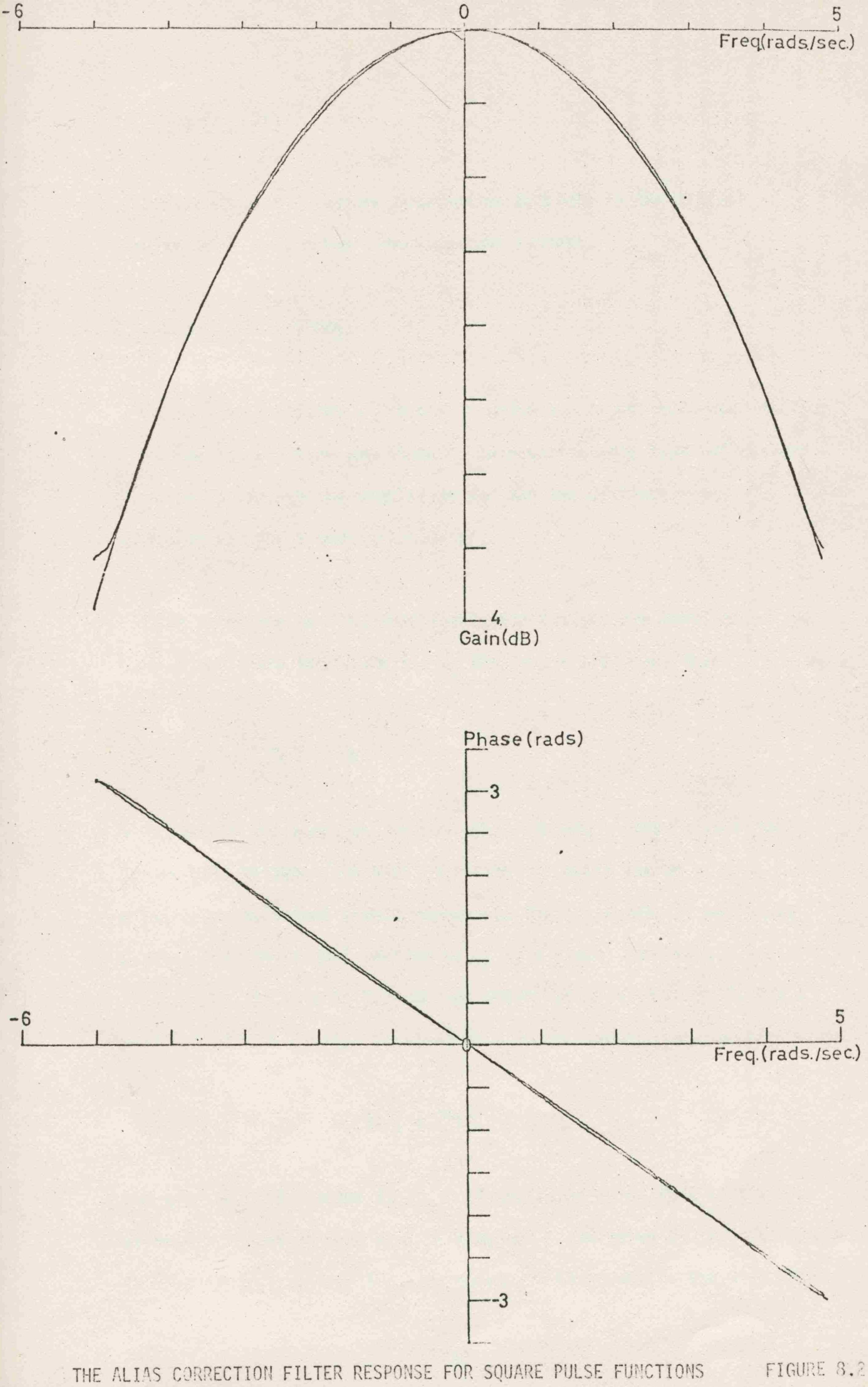


Fig. 8.1 The Complex Oscillator



THE ALIAS CORRECTION FILTER RESPONSE FOR SQUARE PULSE FUNCTIONS FIGURE 8.2

9. SUNDRY ITEMS

This chapter discusses outstanding problems in the digital simulation of continuous communications systems.

9.1 Modulation Systems

Modulation defines a process in which one signal acts upon another to change its phase or amplitude. In practice many types of devices are used to perform the modulation and the two distinct areas of amplitude and phase modulation exist.

It is common to find that modulation systems are bandlimited and that if ω_b is the bandwidth and ω_c the centre frequency then:

$$\omega_c \gg \omega_b. \qquad 9.1$$

Such systems are known as 'narrow-band' systems. When simulating narrow-band systems 'low-pass equivalent' signals can be used. A low-pass equivalent signal represents the instantaneous amplitude and phase of the signal centred at ω_c by a signal centred at zero frequency. No loss of information occurs in this process if, for a narrow-band signal ($f(\omega)$ (centred at ω_c)) the low-pass equivalent is:

$$g(\omega) = f(\omega)\, e^{-j\omega_c t}, \qquad 9.2$$

and $g(\omega)$ is band-limited to ω_b. It should be noted that if $f(\omega)$ is asymmetric about ω_c then $g(\omega)$ is complex. The value of this process is that if $g(\omega)$ is used then the necessary bandwidth of the simulation is much less than before. Thus, if low-pass equivalent simulation is used, much less processing is involved in implementing the simulation. This technique is reported and investigated in References 10, 12 and 40.

9.1.1 Amplitude Modulation

Fundamentally amplitude modulation involves the multiplication of two signals. The modulated signal is therefore:

$$f(t) = a(t)\ b(t). \qquad 9.3$$

Now, if a(t) is band-limited to ω_a and b(t) to ω_b then the bandwidth of f(t) is:

$$\omega_f = \omega_a + \omega_b. \qquad 9.4$$

The problem here with respect to digital simulation, is that the bandwidth increase of Equation 9.4 may produce alias errors and thus appropriate band-limiting safety precautions will be needed.

The system defined by Equation 9.3 is ideal. In practice various methods are used. These methods can be considered in two fundamental ways.

(i) Multiplicative modulation, as Equation 9.3.
In this case the abovementioned considerations are necessary for accurate simulation. For narrow-band systems only the in-band parts of the signal need be considered, this often means modulation by a simple sinusoid only. (For example in the case of switching

modulators only the fundamental of the effective multiplying square wave need be used in a narrow-band simulation).

(ii) Non-linear device modulators apply a non-linear transfer characteristic to the sum of the two applied signals. Non-linear devices are discussed in Section 9.2.

9.1.2 Angle Modulation

In angle modulation systems a signal is used to alter the instantaneous phase of a sinusoidal signal generator. That is:

$$f(t) = \cos(\omega_0 t + \theta(t)) \qquad 9.5$$

or more generally:

$$f(t) = e^{j(\omega_0 t+\theta(t))} \qquad 9.6$$

The bandwidth of f(t) will be wider than that of θ(t) due to the well known relationships of angle modulators, and thus appropriate band-limiting will be necessary. Often such systems will be narrow-band. Where low-pass equivalent simulation is appropriate:

$$g(t) = e^{j\theta(t)}. \qquad 9.7$$

9.1.3 Amplitude Demodulation (Detection)

The process of demodulating an amplitude modulated signal is ideally the same as the modulation process. The demodulation involving the multiplication of two signals. Therefore the discussion of amplitude modulation is also relevant to idealised detection.

Practical amplitude demodulation is often achieved by devices which detect the envelope of the modulated signal with respect to a centre frequency. The majority of such systems are narrow-band and therefore 'low-pass equivalent' simulation can be used. Consider the low-pass signal:

$$a(t) = u(t) + jv(t), \qquad 9.8$$

which represents a narrow-band signal centred at ω_o.
The narrow-band signal is thus:

$$f(t) = a(t)e^{j\omega_o t} + a^*(t)e^{-j\omega_o t}, \qquad 9.9$$

where '*' denotes complex conjugate. Thus:

$$f(t) = 2u(t) \cos \omega_o t + 2v(t) \sin \omega_o t, \qquad 9.10$$

and considering the demodulating signal:

$$b(t) = \cos \omega_o t, \qquad 9.11$$

then the demodulated signal is:

$$g(t) = f(t) \cos \omega_o t. \qquad 9.12$$

The low frequency content of g(t) is:

$$c(t) = u(t). \qquad 9.13$$

Therefore where 'low-pass equivalent' simulation is used the low-pass demodulated signal is the real part of the complex equivalent signal.

9.1.4 Angle Demodulation

Consider the generalised signal:

$$f(t) = e^{(a(t) + j\phi(t))} \qquad 9.14$$

then the phase information is:

$$\phi(t) = \mathrm{Im}\{\ln f(t)\} \qquad 9.15$$

or (for frequency demodulation) the instantaneous frequency is:

$$\omega_i(t) = \frac{d}{dt}\phi(t), \qquad 9.16$$

which for:

$$f(t) = u(t) + jv(t), \qquad 9.17$$

becomes,

$$\omega_i(t) = \left\{\frac{u(t)\,\dot{v}(t) - \dot{u}(t)\,v(t)}{u^2(t) + v^2(t)}\right\}. \qquad 9.18$$

These relationships describe the ideal case for a 'low-pass equivalent' signal. In practical cases simulation will need to be considered in other ways. For instance, the 'Foster-Seely' discriminator can be considered as two band-pass filters and two envelope detectors, and could therefore be simulated as such (Reference 40).

9.2 Non-linear Devices

The types of non-linear devices considered were introduced in Section 1.1.4. In this introduction the basic properties of 'soft' and 'hard' characteristics were defined.

9.2.1 'Soft' Characteristics

A 'soft' characteristic is defined by a power series relationship, that is, for input x(t) and output y(t):

$$y(t) = \sum_{i=0}^{N} a_i \{x(t)\}^i . \qquad 9.19$$

Now if x(t) is band-limited to ω_x then the band-limit of y(t) is:

$$\omega_y = N\,\omega_x. \qquad 9.20$$

Therefore for sampled signals to accurately represent y(t):

$$\omega_x = \frac{\omega_s}{2N}, \qquad 9.21$$

where ω_s is the radian sampling frequency.

9.2.2 'Hard' Characteristics

Typical 'hard' characteristics are rectification or switching functions. Such functions can be considered as 'soft' characteristics where N (Equation 9.19) is infinite. In practice a_i will tend to zero as N tends to infinity and some practical limit will exist where a_i becomes negligible. This limit will thus define the bandwidth spreading limits and the appropriate band-limiting required. Thus it may be possible to derive a 'hard' characteristic model from a power series simulation.

9.2.3 'Narrow-Band' Simulations

Consider the case where a system is simulated by using 'low-pass equivalent' methods. Thus the simulated signal is:

$$f(t) = a(t)\cos(\omega_0 t - \phi(t)) \qquad 9.22$$

'In-band' components of the resultant non-linear processed signal only exist for odd members of the power series as $\cos^n \omega_0 t$ only has components at a frequency of ω_0 when n is odd. Thus if a constant is used to represent the sum of the odd members of the series, that is:

$$k = \sum_{i=o}^{M} a_{(2_i+1)} , \qquad 9.23$$

then the non-linear result for the low-pass signal is:

$$g(t) = k \sum_{i=o}^{M} \left\{a(t)\, e^{-j\phi(t)}\right\}^{(2_i+1)} \qquad 9.24$$

where $M = \frac{N}{2} - 1$ for N even or,

$M = \frac{N-1}{2}$ for N odd.

Band-limiting precautions are obviously required to allow for the bandwidth increase of the low-pass signal.

9.3 Simulation of Systems with Feedback

One of the advantages of the 'sample-by-sample' approach to digital simulation is that systems with feedback are easily catered for. Consider the system shown in Figure 9.1(a) and its digital simulation shown in Figure 9.1(b). The simulating signal p(nT) is delayed by one sampling interval due to the necessity of calculating the sample values. Thus the system model has a delay error in the feedback loop. The delay can be considered as the insertion of a digital filter in the loop the transfer function of which is:

$$H(Z^{-1}) = Z^{-1} \qquad 9.25$$

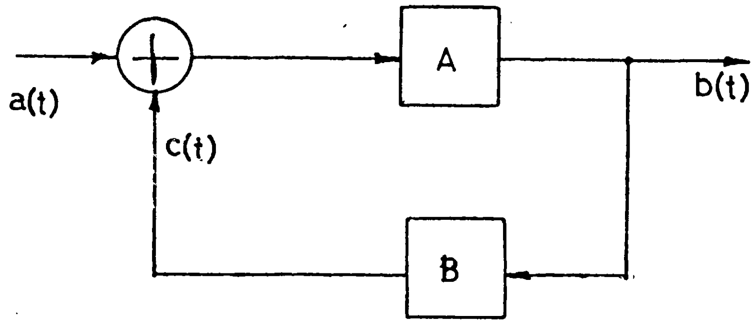
This problem can be overcome if a filtering element already exists in the loop, where the simulating filter has a zero inside the unit circle in the Z^{-1} plane. Such a zero can be removed if it lies at $Z^{-1} = 0$ or inverted if non-zero. That is, for a zero:

$$G(Z^{-1}) = a + Z^{-1}, \qquad 9.26$$

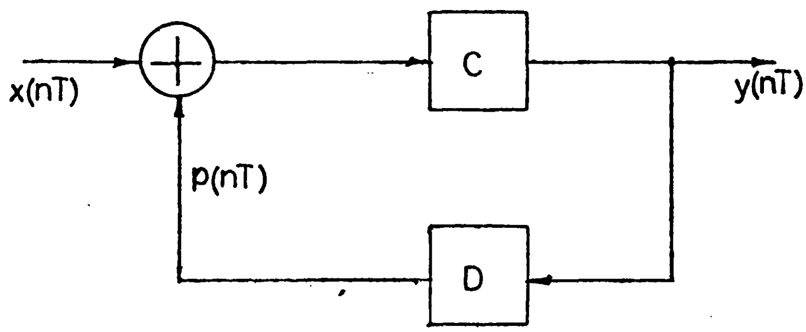
where $|a| < 1$ then a zero can be used such that:

$$G'(Z^{-1}) = 1 + a\,Z^{-1}, \qquad 9.27$$

thus removing the sample calculation delay. (Note: if the filter to be modified is contained in the sub-system 'c' then an additional delay will be needed after the feedback system to provide the correct y(nT).



9.1.(a)



9.1.(b)

Fig. 9.1 Feedback Systems.

10. CONCLUSION

This work has considered the simulation of communications systems by using digital filtering techniques. The objectives of the study were explained in detail in Chapter 1, and the work has proceeded by consideration of the types of element normally found in communications systems. The introductory sections also considered the implementation of generalised simulation schemes and arrived at recommendations for the production of viable simulation structures. The recommendations in this area are largely based upon the observed historical development of the 'analogue computer' simulators.

The majority of the work reported herein has concentrated upon the development of algorithms for digital filter design. The design criterion being the accurate modelling of the frequency responses of continuous filtering elements. A detailed discussion of this work is found in Chapter 7. The results of the digital filter design algorithms show that accurate frequency response models can be produced. Digital filter design methods for frequency response simulation are not found to any extent in the literature. The majority of the cited work is in fact concerned with design methods for 'practical' digital filters (that is, devices to be used in real filtering situations). However it is true that the response fitting methods found in Chapters 5 and 6 are similar to much work in the area of optimal filter design, the main difference being that this work attempts to avoid the often lengthy non-linear optimisation methods by the use of a linear equation fitting method. The basis of the 'zero insert method' is also a different approach to the design of digital filters, in that use is made of the often valuable design base that can be provided by 's' plane to 'Z^{-1}' plane transformations. An interesting aside to the main filtering element study is the production of a practical machine algorithm for the execution of the Weiner-Lee transforms shown in Section 5.3.

The literature investigated on simulation does not consider in depth the problems associated with the modelling of function generation, non-linear and modulation elements. The digital simulation of such elements involves alias errors. Simulation systems in the past overcame the alias problem by increasing the sampling rate until alias errors became negligible. It is the author's contention that a great deal of processing time can be saved by a reduction in sampling rates. Thus a new approach was taken to the simulation of the function generation devices as discussed in Chapter 8. Basically the method involves the generation of a sampled version of the required function and the use of a digital filter to correct the alias error. Section 8.1.2 shows that for a certain class of functions this method can be successfully employed. Other methods of function generator simulation are considered in Chapter 8.

Time considerations have meant that the modelling of modulation, non-linear and certain types of function generation elements has received a somewhat cursory treatment. In this context the discussions of Chapter 9 amount to little more than a statement of the problem. In terms of basic philosophy the simulation of non-linear elements is very interesting. Section 1.4 developed the argument that the modelling requirements of the time and frequency domains are incompatible in terms of digital simulation. The main body of this report has developed simulation methods based upon the accurate modelling of frequency responses.

Non-linear elements are more amenable to simulation in the time, rather than the frequency, domain. However this is not compatible with frequency domain accuracy for low sampling rates. This problem is one requiring a considerable amount of work. The method of increasing the sampling rates to avoid alias problems is costly, but may be the price to be paid for the accurate modelling of non-linear devices. Modulation elements can suffer from similar problems due to the undefinable frequency range extensions which can occur.

It has been shown that digital filtering methods can profitably be used for the simulation of certain elements of the communications systems. Digital filtering methods have considerable advantages in the employment of processor time by enabling low-order simulations and low sampling rates. However such modelling requires considerable computational effort in the model design stage. In practical circumstances a 'trade-off' will exist to achieve time economy.

Much work remains to be done before a viable system for simulation can be produced that will find general use. Following is a brief account of the areas requiring further effort.

(i) The introduction of an error checking method for the filter design algorithms, which is more applicable to generalised simulation usage than that used for algorithmic development.

(ii) The use of phase equalisation methods to further improve filter design methods.

(iii) Investigation of non-linear and modulation system elements with respect to the requirements for system simulation.

(iv) Further investigation of aperiodic and random function generators in those areas not covered by Chapter 8.

(v) The arrangement of the whole into a generalised system simulation facility along the lines suggested in Section 1.3.

ACKNOWLEDGEMENTS

The author wishes to acknowledge the following individuals and institutions who have enabled this work to be completed.

(i) Professor A.J. Eales, Head of the School of Electrical Engineering of the University of Bath for the facilities provided.

(ii) Mr. J.D. Martin for his advice, understanding and patience when acting as project supervisor.

(iii) The U.K. Science Research Council for its financial support under SRC Grant No. B/70/1294.

(iv) The late Dr. A.W. Keen for helpful discussions and ideas.

(v) The postgraduate students and staff of the School of Electrical Engineering of the University of Bath for useful discussions in the course of the work.

(vi) My wife, Hazel, for her understanding and support during the execution of this work.

(vii) Mrs. P.A. Moore for her assistance and typing, enabling the production of the written work.

REFERENCES AND BIBLIOGRAPHY

1. Abraham, R.G. and Collins, M.W., 'Digital Programs for Simulation', Control Engineering, October 1966, pp 93-99.

2. Bartels, R.H. and Golub, G.H., 'Stable Methods for Obtaining the Chebyshev Solution to an Overdetermined Linear System of Equations', Communications of the A.C.M., Vol. 11, No. 6, June 1968, pp 401-406.

3. Bartels, R.H. and Golub, G.H., Algorithm 328, 'Chebyshev Solution to an Overdetermined Linear System', Collected Algorithms from CACM 1967.

4. Bogner, R.E., 'Frequency Sampling Filters - Hilbert Transformers and Resonators', The Bell System Technical Journal, Vol. 48, No. 3, March 1969, pp 501-510.

5. Bond, F.E. and Meyer, H.F., 'Intermodulation Effects in Limiter Amplifier Repeaters', I.E.E.E. Transactions on Communication Technology, Vol. COM-18, No. 2, April 1970, pp 127-135.

6. Boxer, R., 'A Note on Numerical Transform Calculus', Proceedings of the I.R.E., October 1957, pp 1401-1406.

7. Boxer, R. and Thaler, S., 'A Simplified Method of Solving Linear and Non-linear Systems', Proceedings of the I.R.E., January 1956, pp 89-101.

8. Brennan, R.D. and Silberberg, M.Y., 'The System/360 Continuous System Modelling Program', Simulation, December 1968, pp 301-308.

9. Broome, P., 'A Frequency Transformation for Numerical Filters', Proceedings of the I.E.E.E., 1965, pp 326-327.

10. Cappellini, V., 'Digital Filtering with Sampled Spectrum Frequency Shift', Proceedings of the I.E.E.E., February 1969, pp 241-242.

11. Constantinides, A.G., 'Spectral Transformations for Digital Filters', Proceedings of the I.E.E., Vol. 117, No. 8, August 1970, pp 1585-1590.

12. Crystal, Y.H. and Ehrman, L., 'The Design and Applications of Digital Filters with Complex Coefficients', I.E.E.E. Transactions on Audio and Electroacoustics, Vol. AU-16, No. 3, September 1968, pp 315-320.

13. Deczky, A.G., 'Synthesis of Recursive Digital Filters Using the Minimum p-Error Criterion', I.E.E.E. Transactions on Audio and Electroacoustics, October 1972, pp 257-263.

14. Dertouzos, M.L., Kaliski, M.E. and Polzen, K.P., 'On-Line Simulation of Block-Diagram Systems', I.E.E.E. Transactions on Computers, Vol. C-18, No. 4, April 1969, pp 333-342.

15. Desblache, A., 'Simulation of Data Transmission Systems by Means of a Computer', Telecommunication Journal, Vol. 36, 1969, pp 653-657.

16. Elliot, J.B., 'FLAP, a Simplified Simulation System for Time-Share Users', Simulation, July 1972, pp 19-21.

17. Evans, A.G. and Fishl, R., 'Optimal Least Squares Time-Domain Synthesis of Recursive Digital Filters', I.E.E.E. Transactions on Audio and Electroacoustics, February 1973, pp 61-65.

18. Farris, G.J. and Burkhart, L.E., 'The DIAN Digital Simulation Program', Simulation, May 1966, pp 298-304.

19. Freeman, E., et al., 'SYSTID: System Time-Domain Simulation Program', NASA Contract Report 114914, February 1971.

20. Gagné, R.E. and Baxter, D.C., 'Techniques for the Design of Linear Digital Simulations', Simulation, August 1968, pp 87-96.

21. Giese, C., 'State Variable Difference Methods for Digital Simulation', May 1967, pp 263-271.

22. Gold, B. and Rader, C.M., 'Digital Processing of Signals', McGraw-Hill, 1969.

23. Golden, R.M., 'Digital Computer Simulation of Sampled Data Communication Systems, Using the Block Diagram Compiler BLODIB', Bell System Technical Journal, Vol. 45, No. 3, March 1966, p 345.

24. Golden, R.M., 'Digital Filter Synthesis by Sampled-Data Transformation', I.E.E.E. Transactions on Audio and Electroacoustics, Vol. AU-16, No. 3, September 1968, pp 321-329.

25. Guillemin, E.A., 'Theory of Linear Physical Systems', Wiley 1960, pp 536-556.

26. Hamming, R.W., 'Numerical Methods for Scientists and Engineers', McGraw-Hill, 1962.

27. Harnett, R.T., Sansom, F.J. and Warshawsky, L.M., 'MIDAS - An Analog Approach to Digital Computation', Simulation, September, 1964, pp 17-43.

28. Hart, C.E., 'Improved Function Generation Subprograms for Use with CSMP or Other Digital Simulation Programs', Simulation, August 1971, pp 77-79.

29. Holmes, 'The Matched Z Transform', 1970 Digital Filtering Conference, Imperial College, London.

30. Horowitz, J.M., 'Synthesis of Feedback Systems', Academic Press 1963, pp 593-599.

31. ICL, The Manual of the ICL System 4 Analogue Simulation Program.

32. Jury, E.J., 'Theory and Application of the Z Transform', Wiley 1964, pp 24-44.

33. Katz, P., and Bonen, K., 'Digital Simulation of Complex Dynamic Systems', Israel Journal of Technology, Vol. 8, No. 1-2, 1970, pp 163-171.

34. Kelly, R.W. and Hariharan, P.R., 'Ideal Limiting of Periodic Signals in Random Noise', I.E.E.E. Transactions on Aerospace and Electronic Systems, Vol. AES-7, No. 4, July 1971, pp 644-651.

35. King, R.E. and Condon, G.W., 'Frequency-Domain Synthesis of a Class of Optimum Recursive Digital Filters', International Journal of Control, Vol. 17, No. 3, 1973, pp 497-509.

36. Korn, G.A., 'Back to Parallel Processing', Simulation, August 1972, pp 37-45.

37. Kuo, F.F. and Kaiser, J.F., 'System Analysis by Digital Computer'.

38. Lee, Y.W., 'Statistical Theory of Communication', Wiley 1960, pp 481-501.

39. Lynn, P.A., 'Recursive Digital Filters with Linear Phase Characteristics', The Computer Journal, Vol. 15, No. 4, pp 337-342.

40. Manske, R.A., 'Computer Simulation of Narrowband Systems', I.E.E.E. Transactions on Computers, Vol. C-17, No. 4, April 1968, pp 301-308.

41. Martin, J.D. and Metcalfe, J., 'Bandpass Digital Filters', Proceedings of the Conference on Digital Processing of Signals in Communications, Loughborough, 1972, pp 89-99.

42. Miller, J.A., 'Maximally Flat Nonrecursive Digital Filters', Electronics Letters, March 1972, Vol. 8, No. 6, pp 157-158.

43. Orchard, H.J. and Temes, G.C., 'Filter Design Using Transformed Variables', I.E.E.E. Transactions on Circuit Theory, Vol. CT-15 No. 4, December 1968, pp 385-408.

44. Papoulis, A., 'The Fourier Integral and its Applications', McGraw-Hill 1962, pp 192-215.

45. Rizkin, I.K.H., 'A Method for the Digital Simulation of Non-linear Elements in Radio-Engineering Systems', Radio Engineering, Vol. 24, No. 4, 1969, pp 68-72.

46. Sablalash, M., 'A Tellegens Theorem for Digital Filters', I.E.E.E. Transactions on Circuit Theory, January 1971, pp 201-203.

47. Sage, A.P. and Smith, S.L., 'Real-Time Digital Simulation for System Control', Proceedings of the I.E.E.E., Vol. 54, No. 12, December 1966, pp 1802-1812.

48. Sakai, T. and Niimi, Y., 'A Programming System for Block-Diagram Simulation and its Application', Electronics and Communications in Japan, Vol. 15-C, No. 9, 1968, pp 133-142.

49. Saltzberg, B.R., 'Frequency Modulation Using Digital Filtering Techniques', I.E.E.E. Transactions on Communication Technology, Vol. COM-18, No. 5, October, 1970, pp 632-637.

50. Schlechtendahl, E.G., 'DYSYS - A Digital Computer Program for Simulation of General Dynamic Problems', Nuclear Engineering and Design, No. 14, 1970, pp 104-108.

51. Selfridge, R.G., 'Coding a General Purpose Digital Computer to Operate as a Differential Analyser', Proceedings of the 1955 Western Joint Computer Conference.

52. Stearns, S.D., 'Digital Simulation of Frequency-Limited Random Functions', Simulation, January 1969, pp 37-41.

53. Steiglitz, K., 'Computer-Aided Design of Recursive Digital Filters', I.E.E.E. Transactions on Audio and Electroacoustics, Vol. AU-18, No. 2, June 1970, pp 123-129.

54. Storer, J.E., 'Passive Network Synthesis', McGraw-Hill 1957, pp 272-276.

55. Strauss, J.C., et al., 'The SCi Continuous System Simulation Language', Simulation, December 1967, pp 281-303.

56. Su., K.L., 'Time Domain Synthesis of Linear Networks', Prentice-Hall 1971, pp 119-125.

57. Trevor, A.B. and Wait, J.V., 'DARE III B-a CSSL-Type Batch-Mode Simulation Language for CDC 6000-Series Computers', Simulation, June 1972, pp 215-226.

58. Tufts., D.W., Rorabacher, D.W. and Mosier, W.E., 'Designing Simple Effective Digital Filters', I.E.E.E. Transactions on Audio and Electroacoustics, Vol. AU-18, No. 2, June 1970, pp 142-158.

59. Tuttle, D.J., 'Network Synthesis', Vol. 1, Wiley 1958, pp 387-395 and pp 760-767.

60. Veron, M.M., 'Calcul Numérique-Simulation Numérique D'un Systéme Linéaire par une Formulation Récurrente Déduite de la Transformation en Z', C.R. Acad. Sc. Paris t 271, July 1970, pp 100-102.

61. White, G.W.T., 'Methods of Modelling and Simulating Dynamic Systems', Measurement and Control, Vol. 4, August 1971, pp 139-146.

62. Wilts, C.H. 'Principles of Feedback Control', Addison-Wesley 1960, pp 196-201 and pp 261-265.

63. Yang, J.-N., 'On the Normality and Accuracy of Simulated Random Processes', Journal of Sound and Vibration, 1973, pp 417-428.

APPENDIX A - SOFTWARE

The work reported in the main body of this thesis involved a great deal of computer software development. This appendix has been included to give a brief description of the more important items of the software.

The items of software fall into several areas, namely:

(i) routines of general application to the handling of polynomials, linear equations and signal processing;

(ii) routines of general application in the study of digital filtering;

(iii) routines for use in the specific area of simulation filter design, and in simulation studies;

(iv) other routines of interest in the area of signal processing but of no specific application to generalised simulation studies.

The following descriptions of the software will be mostly functional as details of the implementation are felt to be unnecessary in this type of report, and would in any case be too lengthy. In cases where the software was not produced by the author the source will be stated.

A.1. Polynomial and Equation Handling

A.1.1 NEWRA - a routine to find the roots of a polynomial (source - general availability within the School of Electrical Engineering of the University of Bath).

Given the polynomial:

$$H(a) = \sum_{i=0}^{I} h_i a^i \qquad \text{A.1.}$$

then NEWRA calculates the values b_j such that:

$$H(a) \quad = \quad K \prod_{j=1}^{I} (a - b_j) \qquad \text{A.2.}$$

K being a constant.

NEWRA uses the Newton-Raphson iteration procedure from a first order approximation to the Taylor series (see Section 4.3.3). The values of h_i and b_j may be complex.

A.1.2. PARF - a routine to find the partial fraction expansion of a quotient of polynomials $\frac{N(a)}{D(a)}$ with simple poles only.

The first operation is to find the roots b_j of the denominator polynomial, D(a), by the use of NEWRA. Then the coefficients C_j are found by the operation:

$$C_j \quad = \quad \frac{N(b_j)}{\prod_{\ell=1}^{L} (b_j - b'_\ell)} \qquad \text{A.3.}$$

where $\quad b'_\ell \quad = \quad b_\ell$ for $\ell \neq j$

and $\quad\quad = \quad b_j - 1$ for $\ell = j$.

This process is often known as the 'cover-up' rule.

A.1.3. POLMUL - a routine to find the product of two polynomials. That is, given the polynomials D(a) and F(a) then POLMUL finds C(a) such that:

$$\sum_{k=1}^{I+J-1} c_k \, a^{k-1} \quad = \quad \sum_{i=1}^{I} d_i \, a^{i-1} \quad \sum_{j=1}^{J} f_j \, a^{j-1} \quad . \qquad \text{A.4.}$$

In practice this routine was arranged to operate in the 'replacement' mode. That is, after execution the polynomial D(a) becomes the product of the original D(a) and F(a). A flow diagram of a scheme to implement this procedure is shown in Figure A.1.

A.1.4 CALPOL - a 'desk calculator' system for operating upon polynomials.

This program was designed to be used interactively to perform operations on polynomials. The program instructions and data storage are arranged such that the operations are similar to those of a desk calculator.

The data storage consists of seven polynomial stores and two constant number stores. The polynomial coefficients and the constants may be complex. Of the seven polynomial stores three are 'active' stores, forming a stack. These 'active' stores are available to the user for inspection, operation and data entry. The remaining four stores are for temporary use and are accessed via the 'active' stores. The three active stores are denoted X, Y and Z and are best visualised as a stack arranged:

Z
Y
X

The four temporary polynomial stores are denoted A, B, C and D. The two constant numbers are called K1 and K2.

The polynomial X(f) of order I is written:

$$X(f) = \sum_{i=0}^{I} (g_i + j\, h_i)\, f^i \qquad \text{A.5.}$$

the values of g_i and h_i being the values stored in the polynomial stores.

CALPOL is operated by the user entering commands and data from a teletype. The instructions available to the user are denoted by a set of mnemonics of up to six characters. The instructions are as follows, (Note: X' is used here to indicate the state of the X store after execution of the command):

(i)	CLEAR	-	all polynomial coefficients are set to zero.
(ii)	CLEARX	-	X'(f) = 0
(iii)	UP	-	Z'(f) = Y(f)
			Y'(f) = X(f)
(iv)	DOWN	-	Y'(f) = Z(f)
			X'(f) = Y(f)
(v)	XYIN	-	X'(f) = Y(f)
			Y'(f) = X(f)
(vi)	ROLLUP	-	Z'(f) = Y(f)
			Y'(f) = X(f)
			X'(f) = Z(f)
(vii)	ROLLDN	-	Z'(f) = X(f)
			Y'(f) = Z(f)
			X'(f) = Y(f)
(viii)	XFROM	-	X'(f) = A(f) or B(f) or C(f) or D(f)
(ix)	YFROM	-	Y'(f) = A(f) or B(f) or C(f) or D(f)
(x)	XTO	-	A'(f) or B'(f) or C'(f) or D'(f) = X(f)
(xi)	YTO	-	A'(f) or B'(f) or C'(f) or D'(f) = Y(f)
(xii)	ADD	-	Y'(f) = Y(f) + X(f)
(xiii)	SUB	-	Y'(f) = Y(f) - X(f)
(xiv)	MULT	-	Y'(f) = Y(f).X(f)
(xv)	K1*	-	X'(f) = K1.X(f)
(xvi)	K2*	-	X'(f) = K2.X(f)

(xvii) /K1 - $X'(f) = X(f)/K1$

(xviii) /K2 - $X'(f) = X(f)/K2$

(xix) TRAN - $X'(f) = X(K1.f)$

(xx) TRANAD - $Y'(f) = X(f.K_1).Y(f.K_1^*)+X(f.K_1^*).Y(f.K_1)$

$X'(f) = X(f.K_1).X(f.K_1^*)$

where K_1^* represents the complex conjugate of K_1.

(xxi) TRANMU - $Y'(f) = Y(f.K_1).Y(f.K_1^*)$

$X'(f) = X(f.K_1).X(f.K_1^*)$

(xxii) QTRAN - assuming that Y(f) and X(f) form a quotient $\frac{Y(f)}{X(f)}$,

QTRAN performs a transformation such that:

$$\frac{Y'(f)}{X'(f)} = \frac{Y\left\{\frac{P(f)}{Q(f)}\right\}}{X\left\{\frac{P(f)}{Q(f)}\right\}} .$$

(xxiii) X= - enter a polynomial from the teletype into the X store.

(xxiv) K1= - enter a complex constant into K1

(xxv) K2= - enter a complex constant into K2

(xxvi) X - print the coefficients of X(f)

(xxvii) Y - print the coefficients of Y(f)

(xxviii) Z - print the coefficients of Z(f)

(xxix) LOOK - print the coefficients of X(f), Y(f) and Z(f)

(xxx) NORMO - $X'(f) = X(f)/(g_0 + jh_0)$

(xxxi) NORMX - $X'(f) = X(f)/(g_I + jh_I)$

(xxxii) STOP - exit from the CALPOL program.

The CALPOL system operates from a main program and a suite of subroutines. The subroutines and their functions are as follows:

(a) SETA - obtains a list of the command mnemonics

(b) ASRT - obtains and decodes a command mnemonic and prints error messages in the case of illegal entries.

(c) SWOP - performs data transfer operations (iii to xi)

(d) ARITH - performs all arithmetic operations (xii to xviii)

(e) TRAN - performs all constant number transformations (xix to xxi)

(f) QTRAN - performs the QTRAN transformation (xxii)

(g) DAIO - performs all data input/output functions, (xxiii to xxix)

(h) ORCOR - checks the polynomials after the execution of other operations to ensure that the order is correct

(i) POLMUL - a polynomial multiplying subroutine.

Other operations are dealt with by the CALPOL main program.

The CALPOL system operates by acting upon a matrix of polynomial coefficient values, $G_{s,t}$. The value of 't' refers to individual coefficients of the polynomials and 's' defines the polynomial. A vector of seven values U_k us used to indicate which values of 's' refer to which of the polynomials A, B, C, D, X, Y and Z respectively, thus:

$$X_t = G_{s,t}\Big|_{s = U_5} \qquad \text{A.6.}$$

The advantage of such a system is that it is possible to move items (e.g. the XYIN command) by simply changing the values of U_k. A further vector of seven values, V_s, is used to store the order of the polynomials, thus the order of X is:

$$N = V_s\Big|_{s = U_5} \qquad \text{A.7.}$$

A.1.5 POLCAL - a program to find the value of a transfer function expressed as a quotient of polynomials.

Given the transfer function:

$$F(a) = \frac{N(a)}{D(a)} \qquad A.8.$$

then POLCAL calculates:

$$X = F(a)\Big|_{a=b} \qquad A.9.$$

that is:

$$X = \frac{\sum_{i=o}^{I} n_i b^i}{\sum_{j=o}^{J} d_j b^y} \quad . \qquad A.10.$$

A.1.6 SIMU - a routine to solve a set of linear equations. (Source - the University of Bath Computer Unit).

Given a vector d_i and a matrix $a_{i,j}$ then SIMU finds the vector C_j such that:

$$d_i = \sum_{j=1}^{N} a_{i,j} . C_j \qquad A.11.$$

or, written in matrix form, SIMU solves:

$$\begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ \vdots \\ \vdots \\ d_M \end{pmatrix} = \begin{pmatrix} a_{1,1} & a_{1,2} & - - - - & a_{1,N} \\ a_{2,1} & - - - & & \vdots \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ a_{M,1} & - - - - - - & - - - - & d_{M,N} \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ \vdots \\ \vdots \\ C_N \end{pmatrix} \qquad A.12$$

where M = N.   SIMU obtains the solution by a method known as Gaussian elimination.

A.1.7   CHEBY - a routine to find the solution to an overdetermined set of linear equations (Source - Reference 3).

CHEBY finds the solution to a set of overdetermined linear equations, the solution being a Chebyshev or 'mini-max' fit. The equations can be written in matrix form as in Equation A.12, but in this case M > N hence the equations are overdetermined.

A.1.8   FFT - a Fast Fourier Transform program (Source - general availability within the School of Electrical Engineering of the University of Bath).

This program finds the forward or inverse discrete Fourier transform. The FFT is a well known device and therefore no further description is necessary.

A.2. Generalised Digital Filter Software

A.2.1   STAZ - a program to execute the Standard Z Transform of a transfer function in the Laplace variable S. (See Section 2.1.1).

Given the transfer function coefficients p_i and q_n such that:

$$H(s) = \frac{\sum_{i=0}^{I} p_i s^i}{\sum_{n=0}^{J} q_n s^n} \qquad \text{A.13}$$

STAZ first finds the pole positions (r_n) and partial fraction coefficients (c_n) of H(s) (using NEWRA and PARF) such that,

$$H(s) = \sum_{n=1}^{J} \frac{c_n}{s-r_n} \qquad \text{A.14}$$

Then the values r_n' are found:

$$r_n' = e^{jr_nT} \qquad \text{A.15}$$

The denominator coefficients (d_k) of the Z^{-1} plane transfer function are found by:

$$D(Z^{-1}) = \prod_{\ell=1}^{J} (1 - r_\ell' Z^{-1}) \qquad \text{A.16}$$

using the POLMUL subroutine. The numerator of the digital transfer function is calculated from:

$$N(Z^{-1}) = T \sum_{n=1}^{J} c_n \prod_{k=1}^{J} (1 - r_k'' Z^{-1}) \qquad \text{A.17}$$

where:

$$r_k'' = r_k' \text{ for } k \neq n$$
$$= 0 \text{ for } k = n\text{, again using POLMUL.}$$

A.2.2 DIGG1 - a program to find the Matched Z transform.

Given a transfer function defined by a quotient of polynomials:

$$A(s) = \frac{B(s)}{C(s)} \qquad \text{A.18}$$

this program uses NEWRA to find the pole positions g_n and the zero positions f_i such that:

$$A(s) = \frac{K \prod_{i=1}^{I} (s - f_i)}{\prod_{n=1}^{J} (s - g_n)} \qquad \text{A.19}$$

The Matched Z transform (Equation 2.22) is then found, by the use of the POLMUL program,:

$$H(Z^{-1}) = \frac{\prod_{i=1}^{I} (1 - e^{f_i T} Z^{-1})}{\prod_{n=1}^{J} (1 - e^{g_n T} Z^{-1})} \qquad \text{A.20}$$

where T is the sampling interval. The final digital transfer function is then:

$$H'(Z^{-1}) = K\, H(Z^{-1}) \qquad \text{A.21}$$

where the value of the constant 'K' is calculated such that $H'(Z^{-1})$ and A(s) have the same gain magnitude at a given frequency ω', that is:

$$K = \frac{|A(j\omega')|}{|H(e^{-j\omega'})|} \qquad \text{A.22}$$

this value being found by using the POLCAL program.

A.2.3 RESP - a program to obtain values of the frequency response of a transfer function in the Laplace variable 's'.

Given the transfer function defined as a quotient of polynomials, and the radian frequency ω, RESP calculates:

$$X = \frac{\sum_{i=o}^{I} n_i (j\omega)^i}{\sum_{k=o}^{K} d_k (j\omega)^k} \qquad \text{A.23}$$

n_i and d_k being the transfer function numerator and denominator coefficients respectively.

A.2.4 FILCOM - a program to find the frequency responses of a transfer function in the s variable, and a number of transfer functions in the Z^{-1} plane frequency variable, $e^{-j\omega T}$.

FILCOM calculates the frequency responses over a defined frequency range. Within the range the frequency values may be incremented linearly or logarithmically up to the maximum frequency, that is, the Nyquist frequency.

$$\omega_n = \frac{\pi}{T} . \qquad \text{A.24}$$

The lower frequency limit is $-\omega_n$ for linear responses or is defined by the user for logarithmic responses. Thus in the linear case the N frequency values are:

$$\omega_i = \frac{2\, i\, \omega_n}{N} - \omega_n \qquad \text{A.25}$$

$$i = 1,2,3,\text{------}N$$

and in the logarithmic cases the N values are:

$$\omega_{i+1} = \omega_i \cdot \omega' \qquad \text{A.26}$$

$$i = 1,2,3,\text{------}N$$

where

$$\omega' = 10 \left(\frac{\log \omega_n - \log \omega_1}{N}\right) \qquad \text{A.27}$$

ω_1 being the user defined lower frequency limit. Using one of these frequency stepping methods FILCOM is used to compare the responses of a number of digital filter frequency responses with a single continuous frequency response. The continuous response is found by using the RESP program. The digital response values, X_i, are calculated from:

$$X_i = \frac{\sum_{k=0}^{K} a_k \, e^{-jk\omega_i T}}{\sum_{\ell=0}^{L} b_\ell \, e^{-j\ell\omega_i T}} \qquad \text{A.28}$$

a_k and b_ℓ being the digital filter polynomial coefficients. The frequency response values are stored on disc for later plotting.

A.3 Programs for Simulation Filters

A.3.1 DIGG21 - a routine to calculate the position of a pair of inserted zeros as shown in Section 4.4.

DIGG21 uses the Matched Z transform results found by DIGG1 and then calculates the positions of a pair of inserted zeros. The zeros are inserted to correct the Nyquist frequency gain error and to correct the phase response at a frequency specified by the user. This program also corrects the phase response such that the Nyquist frequency digital phase is as close as possible to the continuous phase (see Section 4.2). If the phase correction requires a zero to be placed at $Z^{-1} = -1$ (see Section 4.2) then the routine asks the user if this can be allowed.

A.3.2 DIGG22 - a program to implement the 'root-searching' zero insertion method (see Section 4.3.3).

DIGG22 uses the data provided by DIGG1 to obtain the error function (Equation 4.32). DIGG22 operates by finding the error function slope from closely spaced values around the current estimated root position. The slope value is used to obtain the next estimated root position.

Should the procedure fail to find a root after forty iterations then the search is restarted from a different position. If after a further forty iterations no roots are found then searching is abandoned. The program attempts to find N roots, where N is the number of infinite zeros in the original, s plane, transfer function. Where roots are found with a modulus greater than 100 they are ignored as the effect upon the digital frequency response will be negligible.

A.3.3 DIGG23 - an interactive zero inserter.

DIGG23 uses the results of DIGG1 and plots the responses on a CRT screen. The user may then specify a number of inserted zeros. The results of the zero insertions are then plotted. Thus the user can observe the effects of his zero insertions and make modifications to the zero positions. After several attempts the user can usually produce a frequency response which is a reasonable simulation of the original response. When the user is satisfied with the modified frequency response the program allows the calculation of the final digital transfer function.

A.3.4 Programs for the Implementation of the Unit Pulse Response Method

The following programs are used to implement the simulation filter design method as described in Chapter 5. The design algorithm is described in Section 5.5 and illustrated in Figure 5.1. The individual programs in the suite perform sections of the algorithm, intermediate results being stored on disc. The programs are described with respect to the steps of the flow diagram as shown in Figure 5.1.

(i) IMGEN - first forms the sampled frequency responses of the 'equivalent-bilinear' and 'minimum phase' filters. The program then finds the unit pulse responses of the two filter types and constructs the linear equations. That is, IMGEN executes steps 1 to 4 and 10 to 17 of Figure 5.1. The routines RESP and FFT are used by IMGEN.

(ii) CHEB1 - solves the overdetermined equations of the 'equivalent bilinear' unit pulse response produced by IMGEN using the routine CHEBY. That is, step 5 of the algorithm (Figure 5.1).

(iii) BILPOL - finds the poles and zeros of the transfer function produced by CHEB1 (using NEWRA). The poles and zeros are adjusted to 'de-warped' positions (see Section 2.12), unity magnitude zeros and complementary pole-zero pairs are found and noted for future use. Therefore BILPOL executes steps 6 to 9 of Figure 5.1.

(iv) CHEB2 - solves the overdetermined equations (using CHEBY) of the 'minimum-phase' filter produced by IMGEN. Therefore dealing with step 18 of Figure 5.1.

(v) MINROT - finds the poles and zeros of the transfer function produced by CHEB2. These poles and zeros are checked against those found by BILPOL and if satisfactory any unit magnitude roots are exchanged with those found by BILPOL and the complementary pole-zero pairs are inserted. If the root comparisons are not satisfactory MINROT recommends to the user that the order of the simulation be increased. MINROT thus deals with steps 19 and 20 of Figure 5.1.

(vi) FINAL - uses the results of MINROT and the frequency response produced by IMGEN to invert combinations of zeros to achieve the closest least square phase fit as in step 21 of Figure 5.1. Then, using POLMUL, the final digital simulation transfer function is calculated.

A.3.5 Programs to Implement the 'Direct' Frequency Sampling Method

The programs described in this section implement the simulation filter design method described in Section 6.1.

(i) FREGEN - calculates the data required for Equations 6.2 and 6.4 (using RESP), the frequency samples being taken logarithmically. The number of samples used is organised to be 4 times the expected order of the final simulation filter.

(ii) CHEBF - using CHEBY, CHEBF solves the equations produced by FREGEN and thus gives the final simulation transfer function.

A.3.6 Programs for the Implementation of the 'Indirect' Frequency Sampling Method

The routines described below implement the procedure discussed in Section 6.2 and illustrated by the flow diagram of Figure 6.2.

(i) FREE - a program which uses the RESP routine to calculate 256 equispaced frequency response samples and then uses FFT to implement the Hilbert transform, thus producing the magnitude related minimum phase response. FREE also generates a further set of frequency response samples which are such that they form the frequency response of an equivalent Bilinear Z Transform Digital filter.

(ii) EQGENB - a program to find the overdetermined linear equations (Equations 6.2 and 6.4) of the 'equivalent bilinear' frequency response.

(iii) EQGEN - is similar to EQGENB except that it forms the equations from the 'minimum phase' response.

(iv) SOLVE - a program which uses the CHEBY subroutine to solve the equations produced by EQGEN and EQGENB.

A.3.7 Programs for the Investigation of Correction Filters for Function Generation

The following programs were written to investigate the problems discussed in Section 8.1.2.

(i) PRBF2 - a program to generate the frequency response of the filter specified by Equation 8.27 and subsequently generate the over-determined linear equations such that the filter transfer function may be found. These equations were processed by the SOLVE program to give the filter design of Equation 8.28.

(ii) PRBSIM - a program to test the correction filter designed to correct the alias error of square pulse functions (see Section 8.1.2). The program operates by using the FFT program to spectrum analyse a pseudo-random binary signal which has been passed through the correction filter specified by Equation 8.28. Thus the effectiveness of the correction filter may be assessed.

A.4 Miscellanea

The programs discussed in this section have no direct relevance to the study but they were found to be useful as background information for digital filter studies.

A.4.1 ZEDRES - a program to produce the unit pulse response of a given digital filter.

When supplied with a Z^{-1} plane transfer function ZEDRES constructs the linear difference equation of the transfer function.

This linear difference equation is then excited by a digital unit pulse and thus the output is produced.

A.4.2 - IMPRES - a program to find the sampled impulse response of a filter defined by an s plane transfer function.

IMPRES uses the routines PARF and NEWRA to obtain the partial fraction expansion of the supplied transfer function. Thus, for a transfer function with N poles, the coefficients c_i and a_i are found such that:

$$H(s) = \sum_{i=1}^{N} \frac{c_i}{s + a_i} \qquad \text{A.29}$$

Then, for a sampling interval of T, the impulse response is:

$$h(nT) = \sum_{i=1}^{N} C_i \, e^{-a_i nT} . \qquad \text{A.30}$$

A.4.3 SIM2 - an example of a simulation problem

SIM2 is a program written as a simple simulation of a frequency modulated digital communications system. The transmitter consisted of a fourth order premodulation filter, frequency modulator and an eighth order intermediate frequency filter. The receiving system consisted of an eighth order intermediate frequency filter, a discriminator, a fourth order post-detection filter and a comparator to recover the binary information. Facilities were provided for the addition of adjacent channel interference and transmission medium noise. A schematic of the system is shown in Figure A.2. This program was written for illustrative purposes only and as such was satisfactory.

A.4.4 FILT and DFPAC - real time digital filtering programs.

These programs are based upon the canonic form second order digital filter section (Figure A.3(a)) implementing the transfer function:

$$H(Z^{-1}) = \frac{A + BZ^{-1} + CZ^{-2}}{1 - DZ^{-1} - EZ^{-2}} \quad . \qquad \text{A.31}$$

In practice the implementation is of the form shown in Figure A.3(b). The arithmetic operations of the filter were performed in a two-word floating point format (one word mantissa, one word exponent) thus avoiding the restrictions of fixed point arithmetic and allowing faster implementation than can be achieved using the standard three-word floating point arithmetic provided with the Digital Equipment Corporation PDP-8E computer.

The FILT program allows up to four filter sections (eighth order), each section being separately programmed so as to give high speed. In practice it was found that the PDP-8E computer would allow 0.5mS/section sampling interval when using FILT.

DFPAC is a subroutine implementing a second order filter section. Therefore any filter implementation using DFPAC effectively multiplexes a single second order filter section. The multiplexing action means

means that the execution speed is slower than for the FILT program and in practice 1ms/section sampling interval was the best that could be achieved.

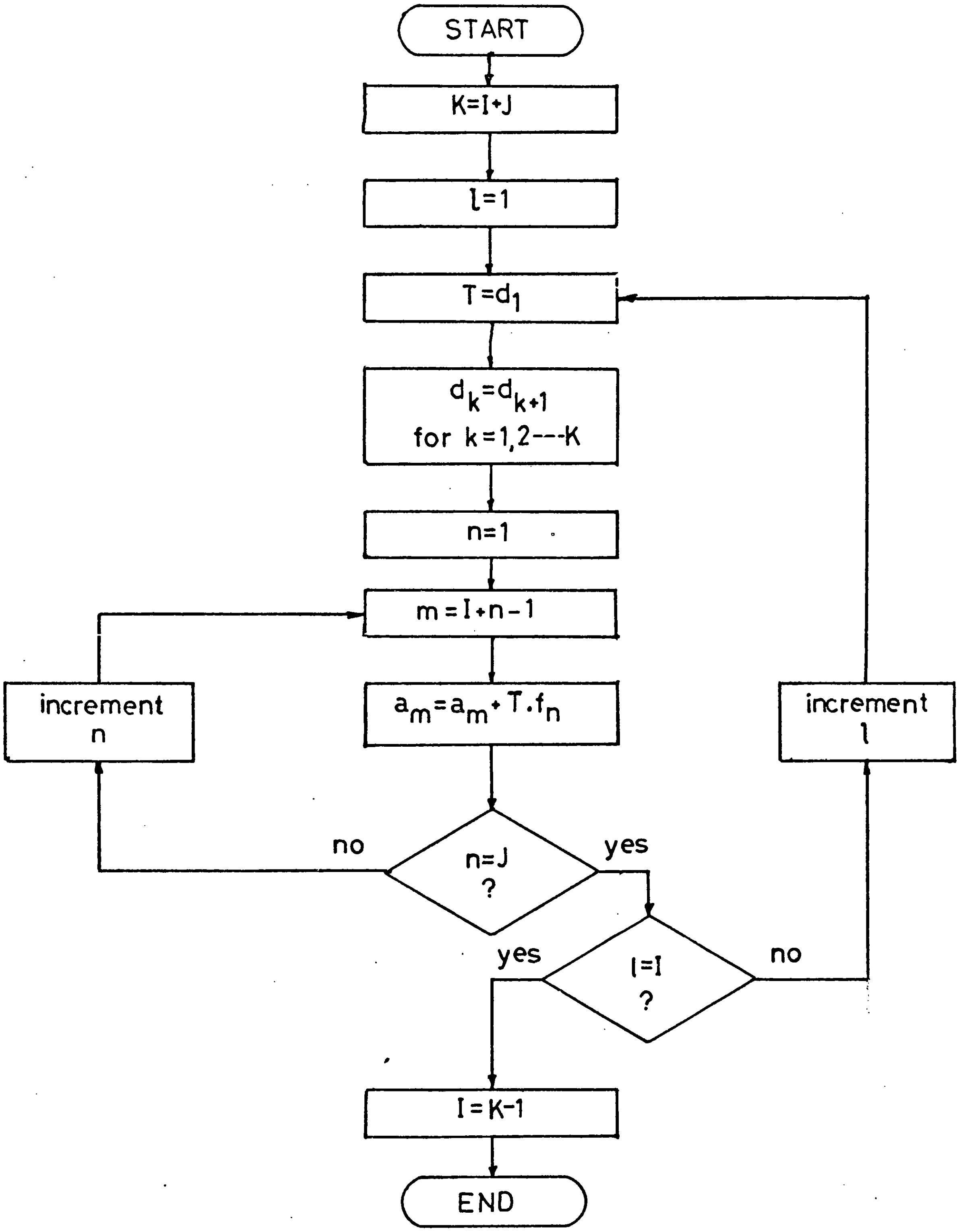


Fig. A.1. The POLMUL program

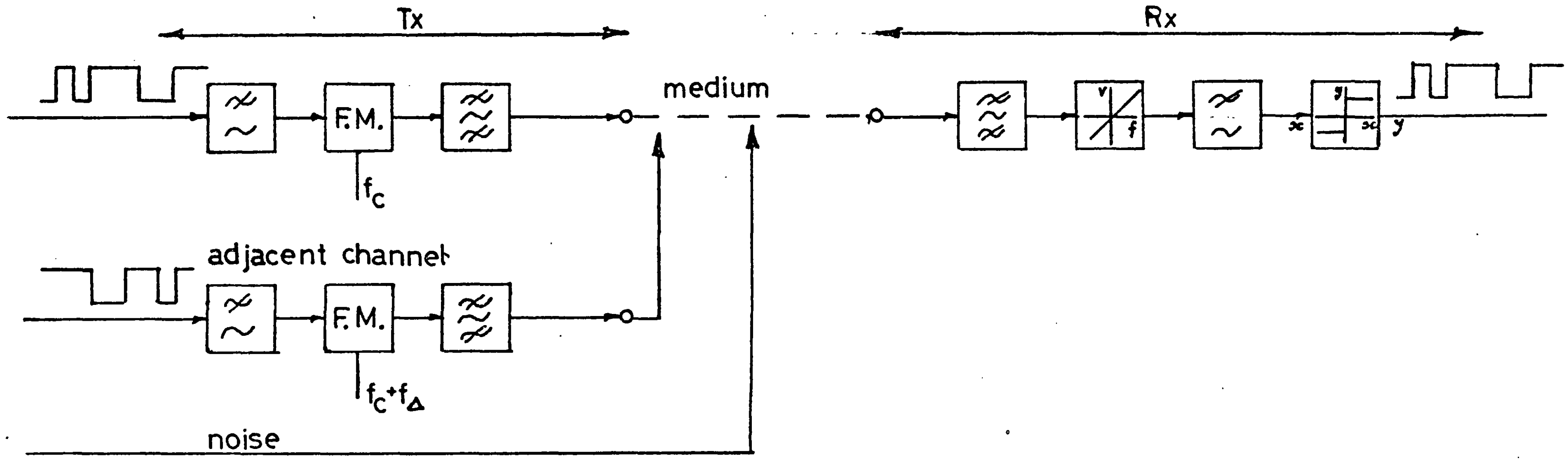


Fig.A2. A Typical Simulation

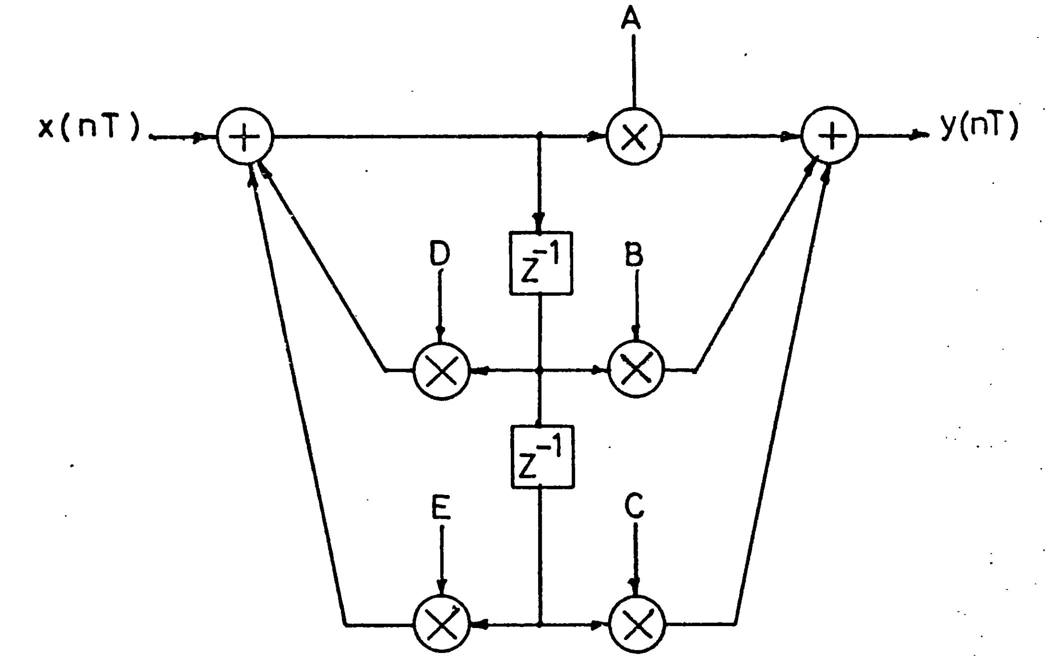


Fig. A.3(a) Schematic

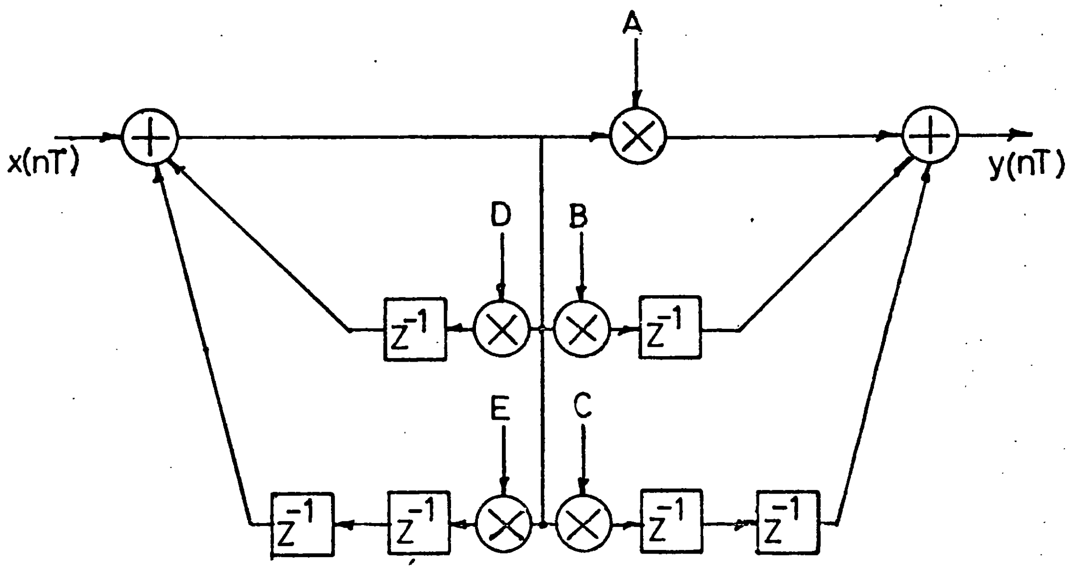


Fig. A.3(b) Implementation

Fig. A.3. The Canonic Form 2^{nd} Order Section

APPENDIX B

A paper presented at the 'Conference on Digital Processing of Signals in Communications', Loughborough, 1972.

BANDPASS DIGITAL FILTERS

J.D.Martin, M.Sc(Eng), C.Eng., M.I.E.E.
and
J.Metcalfe, B.Sc.

Summary:

An investigation is reported into the various methods available of designing digital bandpass filters. Three basic methods are described, each of which can be used with one of three transformations. Experimental results help to spotlight the basic differences between the methods, and conclusions are drawn as to the utility of each.

1. Introduction

Bandpass digital filters are used for simulation of continuous systems, for real-time filtering of sampled signals, and for estimation of power spectra. There are several approaches to the design of a bandpass filter, and it is not immediately clear from the literature as to which method will be best for any particular duty. This short paper sets out the results of an initial investigation, which it is hoped, will enable one to make a more informed judgement on the different methods of design. Three possible techniques are:

a) Bandpass design in the s-plane, followed by a transformation into the z-plane.
b) Lowpass design in the z-domain, followed by a lowpass-bandpass transformation in the z-plane.
c) Lowpass design in the z-plane, but used with a complex modulation of the signal.

The methods are first briefly reviewed, and then the results presented.

2.1 S-plane design

By taking this approach, the many existing aids to filter design can be directly used to design a suitable bandpass filter in the s-domain. However, there are at least three different transformations from the s-plane to the z-plane, and the performance of the final digital filter depends on the type of transformation which is employed.

The standard z-transform approach makes the substitution:

$$\frac{A}{s + \alpha} \Rightarrow \frac{AT}{1 - (\exp -\alpha T) . z^{-1}} \qquad \text{---(1)}$$

into the transfer function expressed as a sum of such terms. The major problems encountered are those of aliasing due to the repeated nature of the frequency spectrum, and the addition to the filter function of unspecified zeros.

* School of Electrical Engineering, University of Bath, Claverton Down, Bath, BA2 7AY, England.

The bilinear z-transform is rather more subtle, making the direct substitution:

$$s \Rightarrow \frac{2}{T}.\frac{1 - z^{-1}}{1 + z^{-1}} \qquad \text{---(2)}$$

Since this transformation maps the whole of the $j\omega$ frequency axis onto the unit circle in the z-plane, aliasing effects have been removed completely. In their place, the frequency scale is warped according to a function of the original frequency variable,

$$\omega' = \frac{2}{T}.\tan^{-1}\frac{\omega T}{2} \qquad \text{---(3)}$$

Thus, the response of a digital filter designed via this transformation cannot correspond exactly with the response of the original continuous filter.

The matched z-transform keeps the original pole and zero positions of the prototype continuous filter, by making the substitution:

$$s + \alpha \Rightarrow 1 - (\exp{-\alpha T}).z^{-1} \qquad \text{---(4)}$$

into the response function expressed as a ratio of polynomials in s. Such a transformation is still subject to aliasing errors, but no additional zeros are added. The frequency scale is still linear.

2.2 Transformation in the z-plane

The bandpass filter resulting from this technique, is transformed in the z-plane from a prototype digital lowpass filter. A lot of design data is now available for lowpass filters, so this is a very attractive idea. There are two principle ways of effecting this transformation. The simplest is to use a linear frequency translation, as described by Broome[1]. Thus the bandpass filter, designed from a lowpass response H(z), is given by:

$$H(z) = H(z.\exp - j\omega_o T) + H(z.\exp j\omega_o T) \qquad \text{---(5)}$$

Two substitutions are necessary, followed by summation of the two responses in the z-domain, giving a ratio of polynomials in z with coefficients modified by $\cos(r\omega_o T)$. Notice the following points:

a) There is the advantage that the frequency scale is still linear, and aliasing errors are avoided.
b) Since the design method corresponds to a frequency translation, bandpass filters can be designed to fall arbitrarily close to the Nyquist frequency, without alias distortion. There will however, be some distortion of the response under such conditions, because the filter image is very near to the proper filter.
c) The bandpass filter characteristic will be symmetrical about its centre frequency, and its bandwidth is the same as that of the lowpass prototype.

If the input signal which is to be filtered is real, then this form of translation must be used, but if an analytic input signal is used, then an analytic filter response can be produced for filtering, which makes use of only one of the terms in equation 5, and results in complex filter coefficients. There is some computational advantage in this approach. (See Crystal et al[2]).

A further technique has been proposed by Constantinides[3], which makes use of a transformation of the form:

$$z^{-1} \Rightarrow -\frac{z^{-2} - \alpha.z^{-1} + b}{bz^{-2} - \alpha z^{-1} + 1} \qquad \text{---(6)}$$

Such a transformation has certain advantages over the previous method of linear translation, in that there is provision for scaling the bandwidth of the filter within the transformation, so that a prototype of any bandwidth can be employed. This transformation also preserves the filter attenuation at certain critical points, such as at the Nyquist frequency, and hence eliminates alias errors. However, the prototype filter response is warped by the transformation.

2.3 Complex signal modulation

The methods above have this in common, that given an nth order prototype, a bandpass filter with 2n poles is produced by the transformation technique. For another filter of a different bandwidth, the calculation must be repeated. An alternative approach is therefore to retain the lowpass prototype filter, and frequency translate the signal into this appropriate frequency range. (Figure 1). A complex modulation ($\exp j\omega_o t$) must be used, so as to preserve any asymmetry in the input signal spectrum. A set of bandpass filters for spectral power analysis can then be realised with a single lowpass filter, and a series of modulators. Apart from the modulation process which will be discussed later, there is an advantage due to the simpler form of the filter in the z-domain.

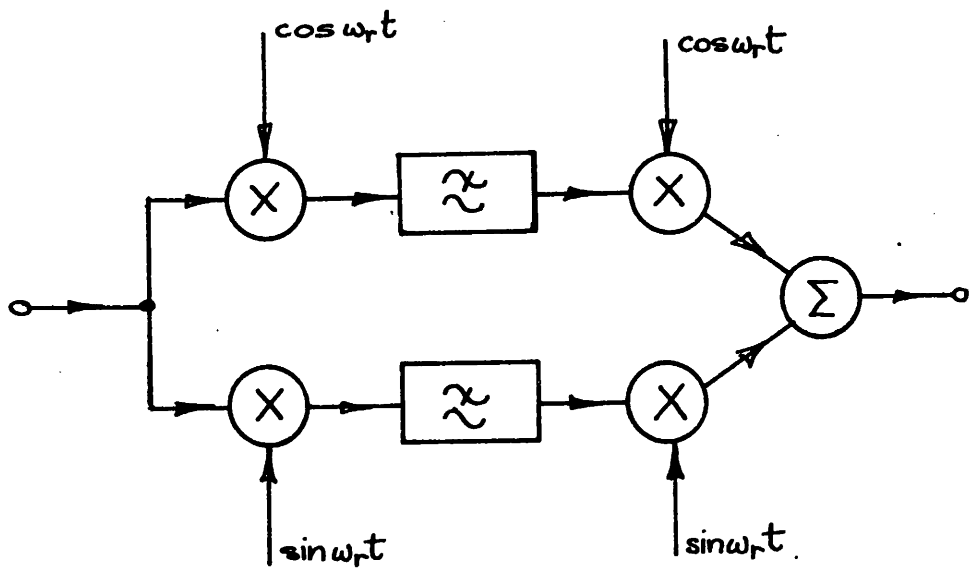


FIGURE 1. COMPLEX MODULATION FILTER

DESIGN OR IMPLEMENTING METHOD		ATTENUATION ERROR AT CUT-OFF FREQUENCIES		ATTENUATION ERROR AWAY FROM PASSBAND		PHASE ERROR AT CUT-OFF FREQUENCIES	
Method	Transform Type	f_1(lower)	f_2(upper)	$0.5 \times f_1$	$2 \times f_2$	f_1	f_2
1. Direct Design from 'S' Plane	Standard	0 dB	0 dB	0 dB	0 dB	3°	3°
	Bilinear	2 dB	-6 dB	0 dB	-6 dB	90°	40°
	Matched	0 dB	0 dB	0 dB	2 dB	50°	60°
2. Design by Unit Function Transforms (6)	Standard	0 dB	0 dB	31 dB	32 dB	10°	5°
	Bilinear	0 dB	0 dB	0 dB	-3 dB	10°	10°
	Matched	0 dB	0 dB	42 dB	42 dB	10°	40°
3. Design by Spectral Shift Transforms (5)	Standard	-1 dB	0 dB	10 dB	-7 dB	2°	2°
	Bilinear	-1 dB	1 dB	10 dB	-8 dB	2°	3°
	Matched	-1 dB	0 dB	11 dB	-7 dB	5°	1°
4. Implementation by Complex Modulation	Standard	0 dB	0 dB	10 dB	0 dB	0°	7°
	Bilinear	0 dB	0 dB	10 dB	-1 dB	0°	7°
	Matched	0 dB	0 dB	11 dB	0 dB	2°	9°

NOTE: Attenuation error is continuous-case attenuation, less the digital-case attenuation.

TABLE 1. Comparison of digital bandpass filters with a continuous bandpass filter.

The actual filter is lowpass, which therefore has n poles instead of 2n poles for the equivalent bandpass filter. However, the signal is complex because of the complex modulation operation, which means that the lowpass filter has to be multiplexed in order to filter both real and imaginary parts of the signal. A similar number of computations are carried out as in the normal bandpass case(2), but here they are related to a lowpass filter and so could be expected to be less sensitive to pole errors than the equivalent bandpass filter. If the input signal is known to be only amplitude modulated, then only the real part of the signal need be processed, giving a processing gain of two.

The modulator is an additional element, which does increase the complexity of the overall filter. For a relatively high signal centre frequency, the modulator may need to be of an analog type anyway, but the digital sampling clock can be used to synchronise the modulator frequency(4). For cases where the total input signal bandwidth is within the Nyquist frequency, for instance in spectral power estimation, a digital modulator can be used, perhaps with provision to multiplex the modulator frequency over the necessary frequency band. Signals of the form $\cos(n\omega_r T)$ and $\sin(n\omega_r T)$ are required. (T is the sampling interval, ω_r is the desired filter centre frequency, and n is the discrete time variable). The modulator signal frequency may be derived by a digital oscillator as described by Saltzberg(5), or alternatively from accessing a sine/cosine table contained in a read-only-memory. Such ROMs are now becoming readily available. Certain special cases can be identified, in which particularly simple modulating functions can be used (see Tufts et al(6), and Capellini(7), who takes the special case of $\omega_r = \omega_s/2$ giving a set of bandpass filters, which divide the available frequency axis into 2^m equal bands.

3. Practical results

Filters have been designed to the following specification by the methods above.

Type:	3rd order Butterworth bandpass response.
Bandwidth:	0.1 rad/Sec.
Centre frequency:	1 rad/Sec.
Sampling frequency:	10 rad/Sec.

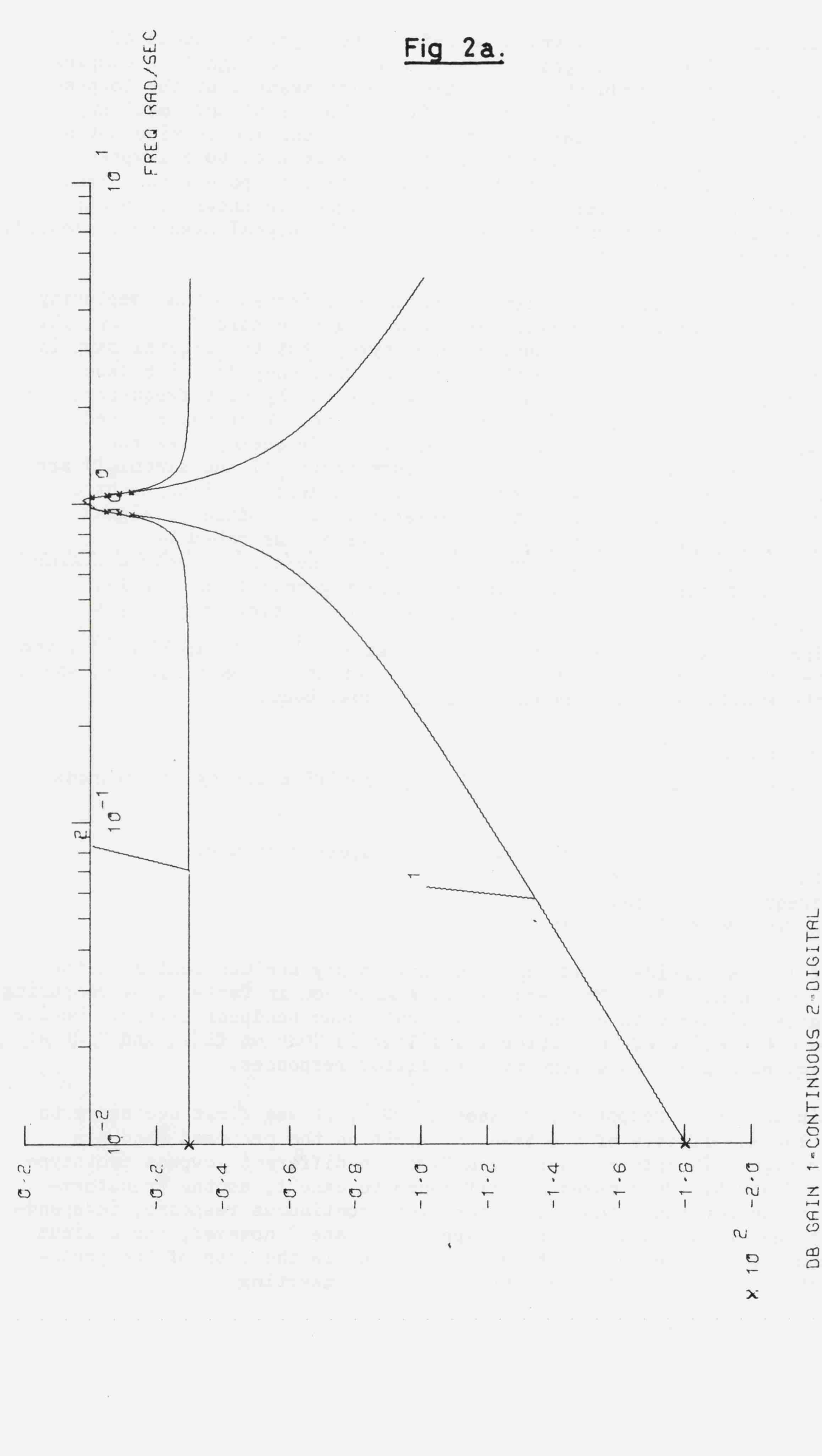
The Q of 10 was considered adequate to show up any serious faults in the various design methods. The results are summarised in Table 1, by comparing each digital filter with a conventional continuous bandpass filter. Notice that the attenuation of the comparison filter is 70db at $f1/2$, and 71db at $2f_2$. Figures 2 and 3 show some typical filter responses.

To obtain the above response for cases 3 and 4, it was first necessary to produce lowpass filters of the same bandwidth as the proposed bandpass filter design. Therefore cases 1 and 2 used a different lowpass prototype to cases 3 and 4. This makes no difference to case 1, as the transformations in the s-plane produce the same final continuous response, independently of the bandwidth of the prototype. For case 2 however, the maximum stop band loss of the resultant bandpass filter is the loss of the prototype filter at the Nyquist frequency. That is inserting

$$z = 1 \ (\omega = 0)$$

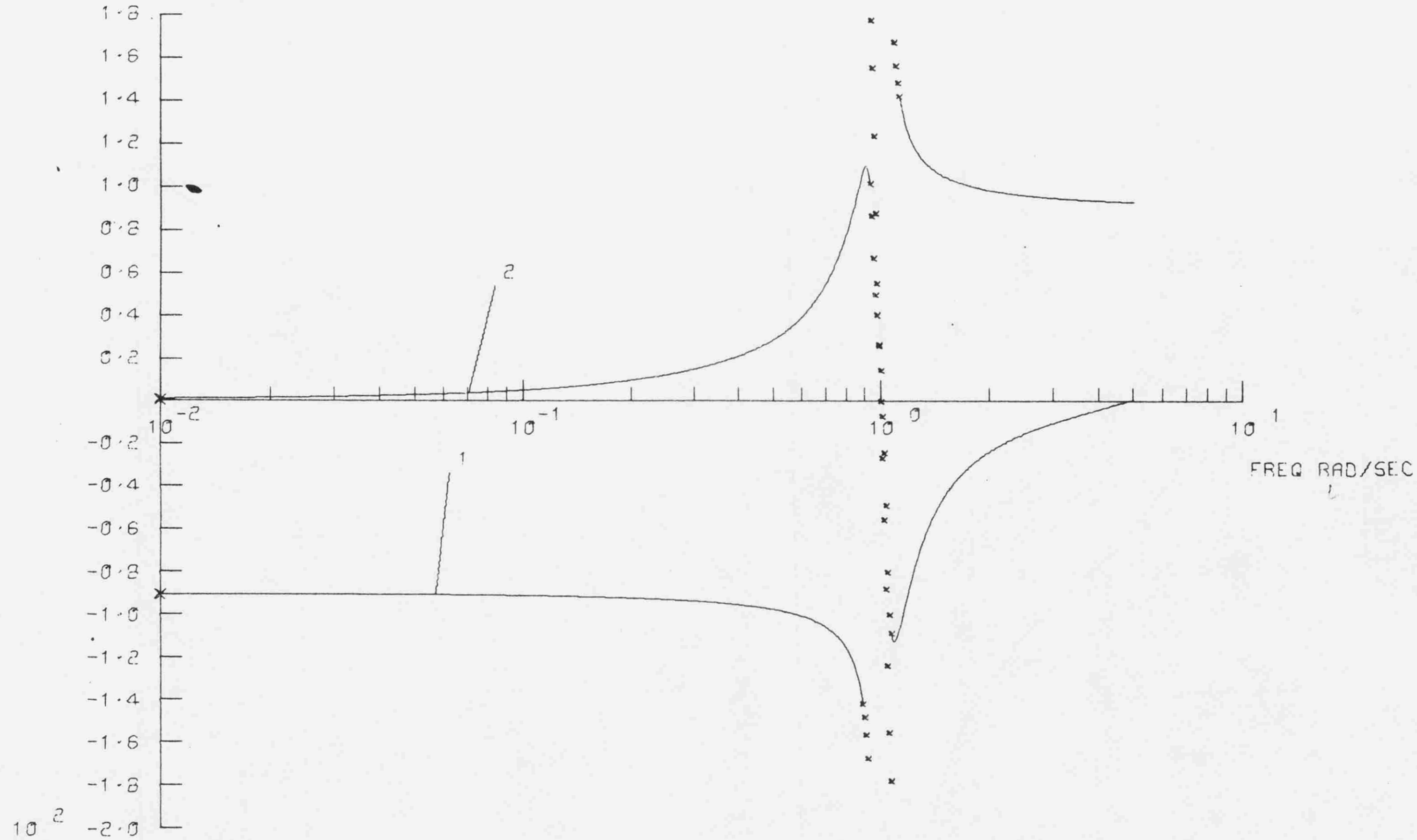
and $$z = -1 \ (\omega = \omega_s/2)$$

Fig 2a.



FILTER COMPARISON (Case 2, by Matched Z Transform)

Fig 2b.



PHASE 1-CONTINUOUS 2-DIGITAL

FILTER COMPARISON (Case 2, by Matched Z Transform)

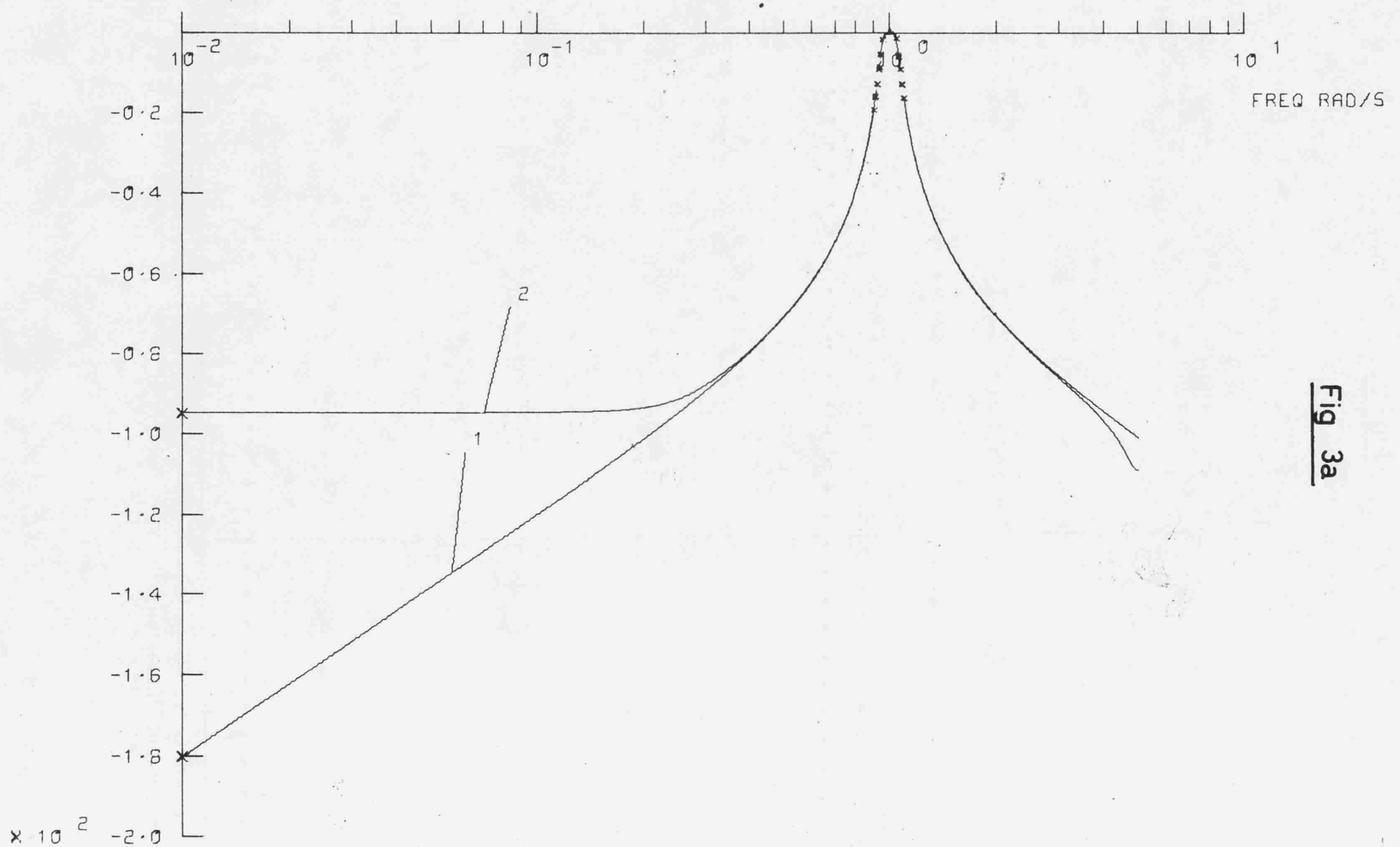
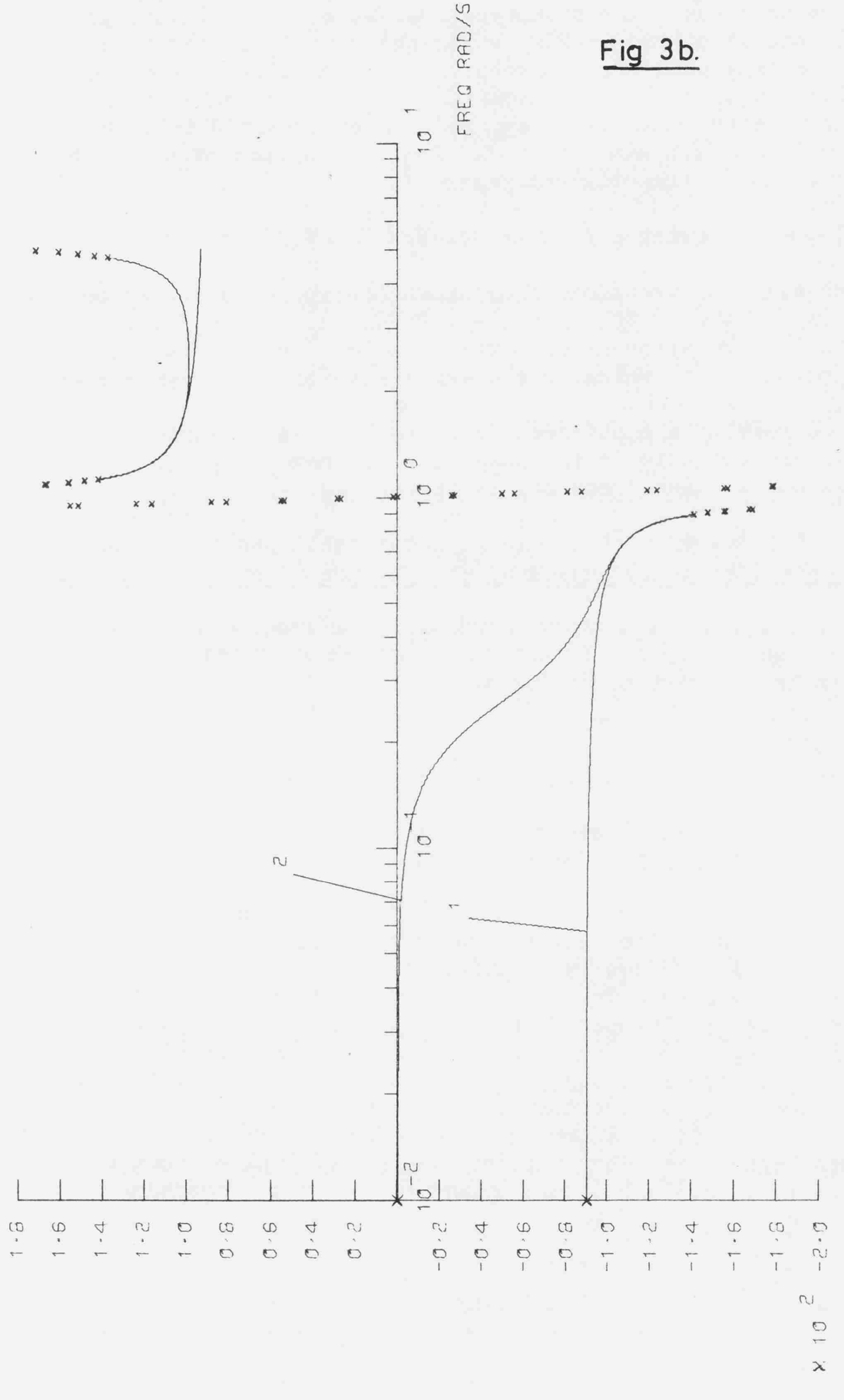


Fig 3a

DB GAIN 1=CONTINUOUS 2=DIGITAL

FILTER COMPARISON (Case1, by Standard Z Transform)

Fig 3b.



FILTER COMPARISON (Case 1, by Standard Z Transform) in equation 6, the transformation equals -1, representing a prototype frequency of $\omega_s/2$. This factor is unimportant when the prototype is designed via the bilinear z-transform, where the loss at $\omega_s/2$ is infinite. However, for designs where the prototype loss at $\omega_s/2$ is finite, the attenuation in the stop band levels out to the prototype loss at $\omega_s/2$. Now the narrower the pass band of the prototype lowpass filter, the greater will be its loss at $\omega_s/2$. So, by implementing this method with narrowband prototypes, much better bandpass responses can be expected than when using wide band prototypes of a similar kind. This limitation of the method is not explained fully in the original paper(3).

From Table 1, the following points are clear:

Method 1. The bilinear transformation shows considerable errors because of the warping of the frequency scale. These may be largely removed by pre-warping the design frequencies. The matched-z transformation shows a considerable phase error, which is characteristic of this transformation.

Method 2. Good results are obtained from the bilinear transformation. The other two are adequate in the passband, but have only limited attenuation as the Nyquist frequency is approached.

Method 3. A satisfactory method, showing some small passband errors. The lower stopband error is caused by the absence of a zero at $z = 1$.

Method 4. Completely satisfactory apart from the same stopband error as method 3. This response is almost identical with that of the continuous filter implemented by a similar method.

4. Conclusions

The chosen method of filter design should satisfy the following criteria:

a) Straightforward initial design technique
b) Acceptable errors in the design method.
c) Economic implementation.

Criterion (a) depends on the initial standard design information which is available. Method 1 requires information on a bandpass continuous filter, which is normally available in system simulation duty. Methods 2,3,4, commence with a digital lowpass filter, and hence are perhaps more suitable for direct filtering duty. Modelling of a bandpass filter at a frequency less than the sampling frequency, often requires that the filter shall have logarithmic symmetry about the centre frequency, so methods 1 and 2 should be used if this property is important. Modelling a filter with a centre frequency much higher than the sampling frequency, implies some kind of signal translation, and logarithmic symmetry is not so important. Methods 3 and 4 could be used here.

If the initial filter information is available in terms of pole positions, then the bilinear and matched transformations offer a simpler substitution technique than the standard z-transform.

For criterion (b), none of the filter design techniques investigated, produce gross errors. For filters of this order, differences in stop band attenuation are not very significant, since the continuous-filter stop-band attenuation is quite good. The worst cases were with method 2, where attenuations of about 30-40db were recorded at the chosen frequencies. The worst pass band errors occurred with the bilinear transformation in method 1, but these are due to warping of the frequency scale and could be eliminated by pre-warping the design frequencies. The bilinear transformation is most useful for cases where good rejection is required. The matched-z transformation should never be used in cases where phase characteristics are important, e.g. for equalisers, or pulse transmission networks.

Methods 3 and 4 have the merit that the frequency is not warped, and neither is there appreciable alias distortion. Method 4, the complex modulation, would appear to be ideal for multiple bandpass filter structures as used for spectal power estimation, provided that a ready method for generating the sine and cosine functions is available. Fewer filter coefficients are required for this technique, but more signal sample storage is needed.

Further comparisons should include the effects of coefficient truncation, which will probably provide another good criterion for deciding between the different methods of design, particularly on the basis of criterion (c). The effect on impulse response of such abberrations as there are, should also have a bearing on the filter performance. It is hoped to investigate these effects further, and report on them at a later date.

5. Acknowledgements

The authors acknowledge the facilities provided at the University of Bath by the Head of the School of Electrical Engineering, Prof.A.J.Eales, and the Head of the Electronics Group, Dr.A.W.Keen. One of the authors (J.M.) gratefully acknowledges the financial support of the Science Research Council.

6. References

1. BROOME, P.: "A frequency transformation for numerical filters", Proc.I.E.E.E., 1965, pp.326-327

2. CRYSTAL,T.H., & EHRMAN,L.: "The design and applications of digital filters with complex coefficients", I.E.E.E.Trans., AU-16, 1968, pp. 315-320.

3. CONSTANTINIDES,A.G.: "Spectal transformations for digital filters", Proc.I.E.E., 117:8, pp.1585-1590, 1970.

4. GLASER,A.B.: "Digital time-division multiplexed N-path system", I.E.E.E.Trans. CT- , pp.600-604, 1970.

5. SALTZBERG,B.R.: "Frequency modulation using digital filtering techniques", I.E.E.E.Trans., COM-18, pp.632-637, 1970.

6. TUFTS,D.W., RORABACHER,D.W. & MOSIER,W.E.: "Designing simple effective digital filters", I.E.E.E.Trans., AU-18, pp.142-158, 1970.

7. CAPPELLINI,V.: "Digital filtering with sampled-signal spectrum frequency shift", Proc.I.E.E.E., 57, pp.241-242, 1969.